

LNCS 2928

Roberto Battiti
Marco Conti
Renato Lo Cigno (Eds.)

Wireless On-Demand Network Systems

First IFIP TC6 Working Conference, WONS 2004
Madonna di Campiglio, Italy, January 2004
Proceedings



IFIP TC6



Springer

Lecture Notes in Computer Science

2928

Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer

Berlin

Heidelberg

New York

Hong Kong

London

Milan

Paris

Tokyo

Roberto Battiti Marco Conti
Renato Lo Cigno (Eds.)

Wireless On-Demand Network Systems

First IFIP TC6 Working Conference, WONS 2004
Madonna di Campiglio, Italy, January 21-23, 2004
Proceedings



Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Roberto Battiti
Renato Lo Cigno
Università di Trento
Dipartimento di Informatica e Telecomunicazioni
Via Sommarive, 14, 38050 Povo, Trento, Italy
E-mail: {battiti, locigno}@dit.unitn.it

Marco Conti
Consiglio Nazionale delle Ricerche (CNR)
Istituto di Informatica e Telematica (IIT)
Via G. Moruzzi, 1, 56124 Pisa, Italy
E-mail: marco.conti@iit.cnr.it

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie;
detailed bibliographic data is available in the Internet at <<http://dnb.ddb.de>>.

CR Subject Classification (1998): C.2, H.4, D.4.4, D.4.6, K.8

ISSN 0302-9743

ISBN 3-540-20790-2 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media
springeronline.com

© 2004 IFIP International Federation for Information Processing, Hofstrasse 3, A-2361 Laxenburg, Austria
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 10967249 06/3142 5 4 3 2 1 0

Preface

This book contains the refereed proceedings of the 1st IFIP-TC6 Working Conference on Wireless-On-Demand Network Systems, WONS 2004. It was sponsored by the IFIP Working Groups 6.3 (*Performance of Computer and Communication Networks*) and 6.8 (*Mobile and Wireless Communications*), and aimed at becoming a premier international forum for discussions between researchers and practitioners interested in the evolution of Wireless Internet Access toward on-demand networking. Ad hoc, routing, localization, resource management, security, applications, performance and analytical models were topics covered in depth by technical papers in this book.

The conference received 77 submissions from 22 countries, showing the worldwide interest. With so many papers to choose from, the Technical Program Committee's job, providing a conference program with the highest quality, was challenging and time consuming. We finally selected 25 full papers for presentation in the conference technical sessions. To give researchers the opportunity to present the novel ideas they are starting to explore, we included in the technical program a poster session devoted to presenting preliminary research results: 7 short papers were selected for presentation in this session. Accepted papers and posters came from 15 different countries.

The technical program also included a keynote speech "Ad Hoc Wireless Networks: Protocols and Applications" by Prof. Mario Gerla, and a panel session devoted to the discussion of the conference topics between academics and industry representatives.

This event would not have been possible without the enthusiasm and hard work of a number of colleagues. A special thanks to the TPC members, and all the referees, for their invaluable help in reviewing the papers for WONS 2004. We would also like to thank all the authors who submitted their papers to this conference for their interest and time, as well as the Springer-Verlag LNCS staff, and Alfred Hofmann in particular, who helped to produce this volume with high standards.

Last, but not least, our very special thanks goes to the people in Trento who made the WONS 2004 event possible, and made this book a reality: Sandro Pera for his continuing organization, Mauro Brunato for solving most of the formatting problems (not only in the book) with creativity, Alessandro Villani for making things work every day and night, Erika Csépi and Leonor Hernández Diaz for managing all WONS-related problems, including those of the colocated events relating to the EURO, INTREPIDO, TANGO, and VICOM research projects, and Elisabetta Nones and Paola Bodio for their logistics expertise.

November 2003

Roberto Battiti
Marco Conti
Renato Lo Cigno

Organization

WONS 2004 was organized by the “Computer Networks and Mobile Systems” Research Program of the Department of Informatics and Telecommunications (DIT) of the University of Trento, Italy, with the help of the Events and Meetings Office of the same university. WONS 2004 was sponsored by IFIP, through the Working Groups 6.3 and 6.8.

Executive Committee

Conference Chair	Roberto Battiti (University of Trento, Italy)
Program Chairs	Marco Conti (IIT-CNR, Pisa, Italy) Renato Lo Cigno (University of Trento, Italy)
Publicity Chair	Mauro Brunato (University of Trento, Italy)
Local Organization Chairs	Sandro Pera (University of Trento, Italy) Alessandro Villani (University of Trento, Italy)
Sponsorship Chair	Paolo Simonetti (Provincia Autonoma di Trento, Italy)

Technical Program Committee

Arup Acharya	IBM T.J. Watson Research Center, USA
Ian Akyildiz	Georgia Institute of Technology, Atlanta, USA
Eitan Altman	INRIA, Sophia Antipolis, France
Roberto Battiti	University of Trento, Italy
Hendrik Berndt	DoCoMo Comm. Laboratories Europe, Germany
Alan Albert Bertossi	University of Bologna, Italy
Giuseppe Bianchi	University of Palermo, Italy
Ernst Biersack	Eurecom, Sophia Antipolis, France
Maurizio Bonuccelli	University of Pisa, Italy
Dragan Boscovic	Motorola Research Center of Paris, France
Andrew T. Campbell	Columbia University, USA
Carla-Fabiana Chiasserini	Politecnico di Torino, Italy
Imrich Chlamtac	University of Trento, Italy
Ananthanarayanan Chockalingam	IIS, Bangalore, India
Sunghyun Choi	Seoul National University, Korea
Marco Conti	IIT-CNR, Pisa, Italy
Sajal K. Das	University of Texas at Arlington, USA
Christos Douligeris	University of Piraeus, Greece
Domenico Ferrari	“Università Cattolica,” Piacenza, Italy
Mario Gerla	UCLA, USA
Silvia Giordano	SUPSI, Lugano, Switzerland
Enrico Gregori	IIT-CNR, Pisa, Italy
Parviz Kermani	IBM T.J. Watson Research Center, USA
Demetres D. Kouvatsos	University of Bradford, UK
Kin K. Leung	Bell Labs, Lucent Technologies, USA
Bo Li	University of Science and Technology, Hong Kong, China
Renato Lo Cigno	University of Trento, Italy
Gerald Q. Maguire Jr.	KTH, Sweden
Petri H. Mähönen	RWTH Aachen, Germany
Francesco Masetti-Placci	Alcatel Research, Paris, France
Michela Meo	Politecnico di Torino, Italy
Sergio Palazzo	University of Catania, Italy
Björn Pehrson	KTH, Sweden
Gian Paolo Rossi	University of Milan, Italy
M. Yahya “Medy” Sanadidi	UCLA, USA
Puneet Sharma	Hewlett-Packard Labs, USA
Ioannis Stavrakakis	University of Athens, Greece
Heinrich J. Stüttgen	NEC Europe Ltd., Germany
Csaba Szabó	BME, Budapest, Hungary
Salvatore Tucci	University of Roma Tor Vergata, Italy
Bernhard Walke	Aachen University of Technology, Germany

Menzo Wentink
Adam Wolisz
Hidetoshi Yokota
Michele Zorzi

Intersil, The Netherlands
TU Berlin, Germany
KDDI R&D Laboratories, Japan
University of Ferrara, Italy

Referees

A. Acharya
I. Akyildiz
E. Altman
G. Auer
I. Awan
L. Badia
V. Baiamonte
R. Battiti
H. Berndt
A.A. Bertossi
E. Biersack
F. Blanchini
Li Bo
L. Bononi
M.A. Bonuccelli
E. Borgia
M. Brunato
R. Bruno
L. Buttyan
A.T. Campbell
C. Casetti
Ling-Jyh Chen
C.F. Chiasserini
A. Chockalingam
M. Conti
F. Cuomo
S. Das
J.C. De Martin
M. Decourville
C. Douligeris
M.J. Fernandez-Getino
Garcia
D. Ferrari
R. Fracchia
M. Garetto

M. Gerla
S. Giordano
F. Granelli
E. Gregori
L. Grieco
S. Hadjiefthymiades
H. Hartenstein
S. Imre
Manhee Jo
A. Kaloxylos
R. Kapoor
P. Kermani
A. Kherani
C. Kiss Kalló
T. Koshimizu
Hong-Yon Lach
P. Laface
Won-Ick Lee
Kin Leung
Bo Li
R. Lo Cigno
G. Maguire
P.H. Mahonen
K. Malhotra
D. Maniezzo
S. Mascolo
G. Maselli
J. Mellor
U. Mengali
M. Meo
P. Michiardi
Q. Ni
K. Oikonomou
E. Pagani
S. Palazzo

A. Panagakis
S. Park
T.R. Park
G. Pau
B. Pehrson
X. Perez-Costa
A. Petrescu
C. Prehofer
G.P. Rossi
S. Salsano
L. Scalia
G. Schembra
R. Schmitz
P. Sharma
H. Stüttgen
K. Sundaramoorthy
C. Szabó
I. Tinnirello
V. Tralli
A. Urpi
A. Villani
E. Viterbo
G. Vivier
B. Walke
R. Wang
M. Wentink
D. Westhoff
A. Wolisz
M. Woodward
K. Xu
H. Yokota
M. Zorzi
A. Zwemmer

Organizing Institutions



UNIVERSITY OF TRENTO - Italy



IFIP

With the financial support of



Table of Contents

Localization and Mobility Management

Markov Localization of Wireless Local Area Network Clients	1
<i>Michael Wallbaum, Torsten Wasch</i>	
A Topology Based Localization in Ad Hoc Mobile Sensor Networks	16
<i>Sridhar Gangadharpalli, Uday Golwelkar, Sridhar Varadarajan</i>	
Grcmob: A Group Mobility Pattern Generator to Evaluate Mobile Ad Hoc Networks Performance	29
<i>Juan-Carlos Cano, Pietro Manzoni, Miguel Sanchez</i>	
Activity-Based User Modeling in Service-Oriented Ad-Hoc-Networks	43
<i>Tobias Breyer, Michael Klein, Philipp Obreiter, Birgitta König-Ries</i>	

MAC and Radio Resource Management

Media Access Control Schemes for Mobile Ad Hoc Networks	57
<i>Chun-Hung Lin, Chien-Yuan Liu</i>	
Throughput Evaluation and Enhancement of TCP Clients in Wi-Fi Hot Spots	73
<i>Raffaele Bruno, Marco Conti, Enrico Gregori</i>	
An Adaptive IEEE 802.11 MAC in Multihop Wireless Ad Hoc Networks Considering Large Interference Range	87
<i>Tzu-Chieh Tsai, Chien-Ming Tu</i>	
A Distributed Algorithm for Bandwidth Allocation in Stable Ad Hoc Networks	101
<i>Claude Chaudet, Isabelle Guérin Lassous, Janez Žerovnik</i>	

Bluetooth Scatternets

Locally Optimal Scatternet Topologies for Bluetooth Ad Hoc Networks	116
<i>Tommaso Melodia, Francesca Cuomo</i>	
An On-Demand Bluetooth Scatternet Formation Algorithm	130
<i>Elena Pagani, Gian Paolo Rossi, Stefano Tebaldi</i>	

Ad Hoc Routing I

GPS-Based Route Discovery Algorithms for On-Demand Routing Protocols in MANETs	144
<i>Mehran Abolhasan, Tadeusz Wysocki</i>	
Dynamic AODV Backup Routing in Dense Mobile Ad-Hoc Networks	158
<i>Wen-Tsuen Chen, Wei-Ting Lee</i>	
Multipath Power Sensitive Routing Protocol for Mobile Ad Hoc Networks	171
<i>Anand Prabhu Subramanian, A.J. Anto, Janani Vasudevan, P. Narayanasamy</i>	

Security, Applications, and Service Support

Dependable and Secure Data Storage in Wireless Ad Hoc Networks: An Assessment of DS ²	184
<i>S. Chessa, R. Di Pietro, P. Maestrini</i>	
A Comparative Analysis on Performance of Mobile IP with Paging Support	199
<i>Hung Tuan Do, Yoshikuni Onozato</i>	
Preventive and Corrective Protection for Mobile Ad Hoc Network Routing Protocols	213
<i>Ricardo Staciarini Puttini, Ludovic Mé, Rafael Timóteo de Sousa Jr.</i>	
Dynamic Service Adaptation for Runtime System Extensions	227
<i>Robert Hirschfeld, Katsuya Kawamura, Hendrik Berndt</i>	

Ad Hoc Routing II

Flood Filtering and Route Selection for Energy-Efficient On-Demand Routing in Wireless Ad Hoc Networks	241
<i>Tran Minh Trung, Seong-Lyun Kim</i>	
High Throughput Route Selection in Multi-rate Ad Hoc Wireless Networks	253
<i>Baruch Awerbuch, David Holmer, Herbert Rubens</i>	
A Three-Tier Framework Supporting Soft QoS Routing in MANET	271
<i>Xin Jin, Hongbo Wang, Yaoxue Zhang, Bin Meng</i>	

MAC Analytical Models

Achieving Maximum Throughput and Service Differentiation by Enhancing the IEEE 802.11 MAC Protocol	285
<i>Bo Li, Roberto Battiti</i>	

Throughput of the Multi-hop Slotted Aloha with Multi-packet Reception	301
<i>M. Coupechoux, T. Lestable, C. Bonnet, V. Kumar</i>	

On Demand Internet Access

WIDE: Wireless Information Delivery Environment in Distributed Hot Spots	315
<i>Mehmet Yunus Donmez, Sinan Isik, Cem Ersoy</i>	

Smart Wireless Access Points for Pervasive Computing	329
<i>Roel Ocampo, Hermann de Meer</i>	

NAT-Based Internet Connectivity for On-Demand Ad Hoc Networks	344
<i>Paal Engelstad, Geir Egeland</i>	

Poster Session

Efficient Management of Domain Foreign Agents in Mobile Computing Environment Using Load Balance	359
<i>Yong Chul Kim, Min Gyo Chung, Jun Hwang</i>	

Node Synchronization Based Redundant Routing for Mobile Ad-Hoc Networks	365
<i>Wonjong Noh, Yunkuk Kim, Sunshin An</i>	

A New Backoff Algorithm to Guarantee Quality of Service over IEEE 802.11 Wireless Local Area Networks	371
<i>Kil-Woong Jang</i>	

A Light Protocol for Distance Estimation in Bluetooth Mobile Ad-Hoc Networks	377
<i>Felipe Gil-Castiñeira, Francisco Javier González-Castaño</i>	

Access Router Discovery and Selection in Advanced Wireless Networks	383
<i>N. Blefari-Melazzi, D. Di Sorte, M. Femminella, L. Piacentini, G. Reali</i>	

Enhancing HIPERLAN/2 Security Aspects	389
<i>Josep L. Ferrer-Gomila, Guillem Femenias, Magdalena Payeras-Capellà</i>	

Analytical Modeling of a Traffic Differentiation Strategy for 802.11 395
 Luca Vollero, Giulio Iannello

Author Index 401

Markov Localization of Wireless Local Area Network Clients

Michael Wallbaum and Torsten Wasch*

Department of Computer Science,
RWTH Aachen University, D-52056 Aachen, Germany

Abstract. Markov localization has been successfully deployed in robotics using highly precise distance sensors to determine the location and pose of mobile robots. In this setting the scheme has shown to be robust and highly accurate. This paper shows how this approach has been adapted to the problem of locating wireless LAN clients in indoor environments using highly fluctuating radio signal strength measurements. A radio propagation model is used to determine the expected signal strength at a given position in order to avoid tedious offline measurements. Some of the issues that had to be addressed include expressing the calculated signal strengths in terms of probability density functions and detecting movement of the mobile terminal solely on the basis of radio measurements. The conducted experiments show that the proposed technique provides a median error of less than 2 m even when there is no line-of-sight to an access point.

1 Introduction

Geolocation systems in general can be based on a dedicated infrastructure or on an infrastructure designed for other purposes. The most popular approach to the latter is to use a wireless communication network. The spread of wireless LANs (WLAN) following the IEEE 802.11 standard [1] has consequently induced research on using the installed equipment for indoor geolocation purposes, as presented e.g. in [2], [3], [4] and [5]. Without modifications to the hardware of commercially available WLAN products such location systems must rely on radio signal strength (RSS) measurements to determine a mobile terminal's (MT) position. The measured RSS is commonly compared to reference values of a so-called radio map determined in the offline phase, i.e. before MT's are actually localized.

The WLAN location determination scheme proposed in this paper is based on the concept of Markov localization by Dieter Fox *et al.* presented in [6]. The idea behind this approach is to use discrete probability distributions to represent the MT's state space. To avoid time-consuming reference measurements the system presented in this paper uses a propagation model to create a radio map. The Markov localization scheme proves to be sufficiently robust to compensate

* The work presented in this paper is supported by the German Federal Ministry of Education and Research under Grant 08NM211.

for the errors introduced by the noisy wireless channel and by the propagation model. For real-world experiments its implementation has been integrated into the WhereMops location service introduced in [11].

The remainder of this paper is organized as follows. Section 2 describes the theoretical background to the proposed scheme, leaving some important – implementation specific – aspects open. These open issues will be tackled in Sect. 3, which introduces a new probabilistic radio propagation model, and Sect. 4, which presents the implementation details of the location determination system. Section 5 describes the evaluation process and the obtained results. Section 6 concludes the paper and gives a preview of possible future work.

2 Markov Localization

2.1 Motivation and Basic Idea

In recent years probabilistic localization schemes such as Markov and Monte-Carlo localization have gained increasing interest in the context of mobile robot systems designed for dynamic environments. Despite the fact that most mobile robots are equipped with rather precise distance sensor equipment deterministic approaches did not cope well in dynamic settings, e.g. with moving people or other objects not incorporated into the world model. Probabilistic approaches inherently model the uncertainties of real-world scenarios and can thus potentially achieve a higher robustness and accuracy.

The problem of dynamic environments also affects WLAN-based location determination systems, however such systems additionally suffer from highly inaccurate “distance sensors”, meaning the measured received signal strength. Radio waves propagate in a very complex manner, the result being that even in a static environment one can observe so-called short term fading, i.e. fluctuations in the received signal strength. Furthermore even small displacements of the MT can significantly alter even the mean of the signal strength readings. The assumption at the outset of the work described in this paper, was that the Markov localization technique would prove sufficiently robust to cope with these adverse conditions, i.e. the dynamic office environment, the inaccurate sensor readings and the inaccurate world model provided by a propagation model, and yet deliver highly precise terminal locations.

Instead of maintaining a single hypothesis as to the MT’s position, Markov localization maintains a probability distribution over the space of all such hypotheses. In the following this probability distribution is called belief. The belief about the MT’s location is only updated when new perceptions are made (i.e. new RSS-measurements have been received) or after the MT has moved. The probabilistic representation allows the scheme to weigh these different hypotheses in a mathematically sound way.

2.2 Notation

Let L_T denote the random variable representing the location of the MT at time $t = T$ and $l = (x, y)^T \in L$ a specific position of the MT within the state space.

Following the notation used in [6] let $Bel(L_T = l)$ denote the probability, that at time $t = T$ the MT is located at a position $l \in L$. In the following the continuous probability distribution is approximated by a discrete, grid-based representation. Positions l that are located outside the grid have a probability of $Bel(L_T = l) = 0$. As $Bel(L_T = l)$ represents a probability distribution the sum over all grid cell values is $\sum_{l \in L} Bel(L_T = l) = 1$.

Finally let $S_T = \langle s_0, s_1, \dots, s_T \rangle$ denote the temporally ordered list of all RSS-measurements and $A_T = \langle a_0, a_1, \dots, a_T \rangle$ the temporally ordered list of all movements conducted by the MT up to time $t = T$.

2.3 Independence Assumptions

This section describes two essential independence assumptions which allow for an efficient recursive implementation of the Markov-Localization algorithm.

Independence of Actions. The state L_T at time $t = T$ solely depends on L_{T-1} and the last conducted action – i.e. movement – a_{T-1} . In other words, all previously reached locations, all sensory input and all previously conducted actions become irrelevant once the current state L_{T-1} is known. This is known as Markov-assumption and is summarized in the following equation:

$$P(L_T = l \mid L_{T-1}, A_{T-1}, S_{T-1}) = P(L_T = l \mid L_{T-1}, a_{T-1}) \quad (1)$$

Independence of Sensor Input. A sensor reading s_T at time $t = T$ solely depends on the state of the environment at $t = T$. Once an MT's state space L_T is known, all previously recorded measurements, states and actions provide no additional information for the calculation of s_t . Equation 2 summarizes this predication.

$$P(s_T \mid L_1, \dots, L_{T-1}, A_{T-1}, S_{T-1}) = P(s_T \mid L_T = l) \quad (2)$$

2.4 The Sensor Model

The sensor model describes how to update the belief about an MT's position $l \in L$ at time $t = T$ given all previously recorded sensor readings S_T . This can be formulated as follows:

$$P(L_T = l \mid S_T) = P(L_T = l \mid s_0, s_1, \dots, s_T). \quad (3)$$

Using Bayes rule and Eq. 2 this can be transformed to

$$= \frac{P(s_T \mid L_T = l) \cdot P(L_T = l \mid S_{T-1})}{P(s_T \mid S_{T-1})}. \quad (4)$$

Obviously the denominator of Eq. 4 is independent of L_T and therefore constant. Furthermore it is assumed that the probability $P(s_T \mid L_T = l)$ for a sensor reading given a certain position is time-invariant. Hence, using $\frac{1}{\alpha_T} = P(s_T \mid S_{T-1})$ and the notation for $Bel(L_T = l)$ introduced at the outset, Eq. 4 can be rewritten as:

$$Bel(L_T = l) = \alpha_T \cdot P(s_T \mid l) \cdot Bel(L_{T-1} = l). \quad (5)$$

This states that the updated belief about the location of an MT upon new sensory input, depends on the probability of the sensory input at a given position weighted by the assumed likelihood of being at this position.

2.5 The Action Model

The belief about the MT's position is not only influenced by the current sensor readings, but also by actions (i.e. movements) of the terminal. Thus there is a need to calculate the probability $P(A) = P(L_T = l \mid A_{T-1})$ that an MT at time $t = T$ is located at position $l \in L$, given all previously conducted movements A_{T-1} . Using the law of total probability this can be written as:

$$P(L_T = l \mid A_{T-1}) = \sum_{l'} P(A \mid B_{l'}) \cdot P(B_{l'}). \quad (6)$$

with $P(B_{l'}) = P(L_{T-1} = l' \mid A_{T-1})$ and $P(A \mid B_{l'}) = P(L_T = l \mid A_{T-1}, L_{T-1} = l')$.

Considering the assumption about independence of actions (Eq. 1) $P(A \mid B_{l'})$ can be simplified such that the probability is only dependent on the last conducted movement a_{T-1} . The term can be further simplified if it is assumed that the probability of reaching a location l given a location l' and an action a_{T-1} is time invariant.

$$P(A \mid B_{l'}) = P(l \mid a_{T-1}, L_{T-1} = l') \quad (7)$$

Resubstituting Eq. 7 in Eq. 6 and using the definition of $Bel(L_T = l)$ the influence of movements on the belief about the MT's location can thus be expressed as:

$$Bel(L_T = l) = \sum_{l'} P(l \mid a_{T-1}, L_{T-1} = l') \cdot Bel(L_{T-1} = l') \quad (8)$$

This states that the probability of being at location l after an action has been performed, can be calculated by summing up the probabilities of reaching l from l' given action a_{T-1} . Each addend is weighted by the likelihood of starting at position l' .

2.6 Algorithm

The previous section has presented the underlying principles of Markov-Localization. Equation 5 provides a recursive scheme for updating the desired density $Bel(L_T = l)$ when new sensor readings s_T are available. Complementary to this Eq. 8 provides the recursive definition of the update procedure when movement of the MT has been detected. The complete algorithm in pseudo-code is shown in listing 2.1.

So far four questions – all of which are highly dependent on the application environment – remain unanswered:

Algorithm 2.1 Markov-Localization

```

1: {initialize probability distribution  $Bel(L_{t=0})$ }
2: loop
3:   if new sensory input  $s_T$  available then
4:     if  $TravelledDistance \geq Threshold$  then
5:       for all Locations  $l$  do
6:         {apply action model}
7:       end for
8:     end if
9:     for all Locations  $l$  do
10:      {apply sensor model}
11:    end for
12:    {normalize resulting distribution}
13:  end if
14:  wait  $\Delta t$ 
15: end loop

```

1. How is $P(s_T | l)$ (Eq. 5) calculated, i.e. the probability of sensor readings depending on the location?
2. How is $P(l | a_T, L_{T-1} = l')$ (Eq. 8) calculated, i.e. the probability of reaching one location from another given a movement?
3. How is movement detected based on RSS measurements?
4. How should the density function for $Bel(L_{t=0})$ be initialized?

All questions are answered in Sect. 4 which describes the implementation details of the Markov localizer. First however the automatic creation of radio maps is discussed, as this is an important prerequisite for the location determination system.

3 Computing Radio Maps

In order to calculate the likelihood of a signal strength measurement given a certain position and base station, the system needs to know what to expect. Radio (signal strength) maps are commonly used to associate reference positions with their expected radio signal strength. The simplest way to build a radio map is by conducting measurements for a set of reference points, with the obvious disadvantage being its enormous costs in terms of time. Trivially the positioning accuracy depends on the distance between the chosen reference points. For example, a desired positioning accuracy of 2 m on one floor of the computer science department building depicted in Fig. 1 would require approximately 300 measurements. Should the floor plan change or an access point be relocated these measurements would have to be repeated. In essence the empirical creation of radio maps is impractical especially considering large-scale deployment of location-based services.

A different approach to generating a radio map given a floor plan is to employ radio propagation models, which are frequently used to plan wireless communication networks. The advantage of calculating instead of measuring the radio

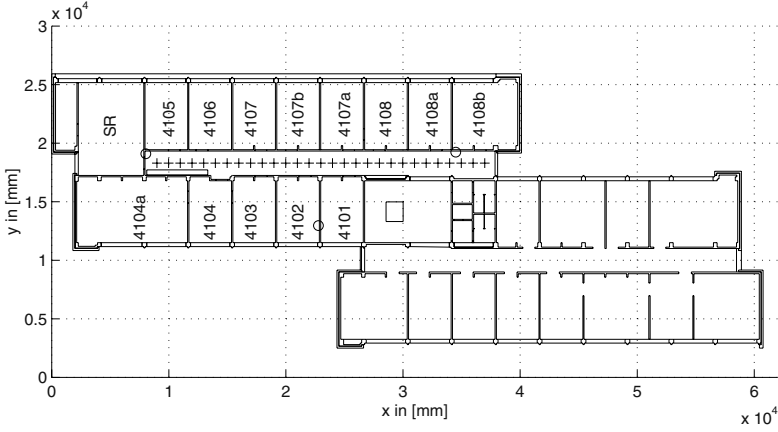


Fig. 1. Floor plan of Chair for Communication and Distributed Systems.

map is less effort and the allowance for arbitrarily fine grained grids. However, radio propagation models can only provide an estimate for the expected signal strengths as radio waves are subject to diverse and complex propagation phenomena. The following section briefly introduces the concept of empirical propagation models which are then modified for the purposes of generating radio maps for Markov localization.

3.1 Empirical Propagation Models

Empirical propagation models are based on the model presented in [8] which was adopted by Seidel und Rappaport to the conditions of indoor environments [9].

$$n_P(d) = n_P(d_0) - 10 \cdot \eta_{FS} \cdot \log_{10} \left(\frac{d}{d_0} \right) - \mathcal{X} \quad (9)$$

The first term on the right side of the equation describes the received signal strength at a reference distance d_0 from the sender under conditions of free-space propagation and is thus an indirect measure of the sender's transmitting power and antenna gain. The unit of n_P is dBm and its value is usually determined by measurements. The second term models the free-space pathloss of the signal. The received signal strength decreases logarithmically with increasing receiver-transmitter displacement d . The parameter η_{FS} is called pathloss exponent and is also estimated by measurements. The last term models the variable attenuation of the signal due to obstructions in the environment.

The RADAR-system [2] employs such a model to calculate a radio map, whereby \mathcal{X} is made a function of the number of walls/obstacles between sender and receiver and an average attenuation for all obstacle types. This model is called Wall Attenuation Factor (WAF) model.

3.2 Design of a Probabilistic Propagation Model

The modifications to the empirical propagation models mentioned above aimed at balancing the accuracy and the effort for estimating the model parameters, by classifying the different obstacle types in the environment. This achieves a higher accuracy than averaging the attenuation over all obstacles as in the WAF-model, yet it requires less effort than the FAF-model described in [10] which requires the attenuation for every specific obstacle.

The novelty of the proposed model is that it calculates an expected value for the RSS at a map location and a measure for the uncertainty or accuracy of the calculated value. These figures can be interpreted as mean and standard deviation of a normal distribution. The calculation of these two values is explained in the following.

Taking a classification of obstacles according to their attenuation properties, let K denote the cardinality of the classification and η_k^{Wall} the attenuation of the k^{th} object class and c_k the maximum number of objects of class k to consider. Using the above definition the expected strength for a position at distance d from an access point given a floor plan can be calculated as

$$n_P(d) = n_P(d_0) - 10 \cdot \eta_{FS} \cdot \log_{10} \left(\frac{d}{d_0} \right) - \sum_{k=1}^K \begin{cases} \mathcal{N}_k \cdot \eta_k^{Wall} & \mathcal{N}_k < c_k \\ c_k \cdot \eta_k^{Wall} & \mathcal{N}_k \geq c_k \end{cases} \quad (10)$$

with

$n_P(d_0)$: signal strength at reference distance d_0 .

$\eta_{FS} \geq 0$: free-space pathloss.

$\eta_k^{Wall} \geq 0$: attenuation of object class k .

\mathcal{N}_k : number of objects of class k between sender and receiver.

c_k : maximum number of considered objects of class k .

Neither the WAF- nor the FAF-model attempt to estimate the deviation of the calculated expected signal strength to corresponding RSS-measurements. However the difference between model and reality is likely to increase with the distance and the number of obstacles between sender and receiver, as the corresponding parameters pathloss exponent and object attenuation can hardly be determined very precisely.

The uncertainty can be modelled as standard deviation of a probability distribution with mean as defined above by $n_P(d)$. To calculate the standard deviation we first define a helper function $\gamma(d) \forall d \geq 0$ which linearly maps the transmitter-receiver separation d to the range $0 \leq \gamma(d) \leq 1$.

For evaluating the location determination scheme proposed in this paper the following two functions modelling the uncertainty have been developed and tested.

$$\sigma_1(d) = 1 + 4 \cdot (1 - \gamma(d))^2 + \frac{1}{2} \cdot \sum_{k=1}^K \mathcal{N}_k \quad (11)$$

and

$$\sigma_2(d) = 1 + 2 \cdot \sqrt{1 - \gamma(d)} + \frac{1}{2} \cdot \sum_{k=1}^K \mathcal{N}_k \quad (12)$$

Both models take into account a standard measurement noise (first addend) and the uncertainty introduced by unprecise attenuation parameters (third addend). The models differ in their second addend which estimates the contribution of the sender-receiver displacement to the error caused by an unprecise pathloss exponent. The effect of creating radio maps for Markov localization on the basis of these functions is described in Sect. 5, where the complete model is also parameterized according to our testbed.

4 Implementation Details

Some issues concerning Markov localization on the basis of RSS measurements have so far not been discussed. This section will address the open questions stated in Sect. 2 and provides some additional implementation details.

4.1 Sensor Model Details

In Sect. 2 it was left open how $P(s_T | l)$ (Eq. 5) should be calculated. Taking the probabilistic propagation model presented in Sect. 3 it is now feasible to determine the likelihood of an RSS reading at a given location. If the expected signal strength calculated for a position and its corresponding uncertainty are interpreted as mean μ and standard deviation σ of a Gaussian distribution then

$$P(a \leq SS_M \leq b) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \int_a^b e^{-\frac{1}{2} \left(\frac{SS_M - SS_C}{\sigma} \right)^2} dSS_M \quad (13)$$

provides an estimate for the probability $P(s_T | l)$ where SS_C denotes the calculated RSS and SS_M the measured signal strength. The integration boundaries have been set to $a = SS_M - \frac{1}{2}$ and $b = SS_M + \frac{1}{2}$.

This probability is calculated for each cell in the probability map associated with a base station for which RSS measurements are available. The resulting maps of all base stations – informally put, each representing a guess concerning the MT’s position based on the resp. information-subset – are then superimposed in a multiplicative manner.

4.2 Movement Model Details

The answer to the question of how to calculate $P(l | a_{T-1}, L_{T-1} = l')$ (Eq. 8), i.e. the probability of reaching one location from another given a movement, has so far been deferred. Before the modelling of movement is tackled it must first be described how movement can be detected in a WLAN without dedicated sensors. At first this presented a major obstacle to the adaptation of Markov localization to WLANs. However it was discovered that the population variance

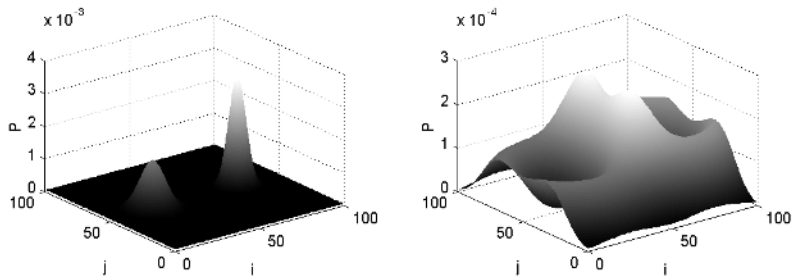


Fig. 2. Belief about MT's position before and after movement model has been applied.

of a sliding mean window, originally used to filter the highly fluctuating raw RSS data, could be used to detect absolute movement quite reliably. Without moving the mobile terminal the variance (of the sliding mean with a window size of 10) seldom exceeded one, even in highly dynamic office environments. When actually moving the mobile terminal the population variance increased to values of six or more at normal walking speed.

The movement model makes use of this phenomenon in order to decide whether the mobile terminal is being moved or not. A population variance above two is interpreted as movement, whereas lower values indicate the MT is stationary. Assuming the mobile terminal has an average speed of $v_{max} = 1.8 \frac{m}{s}$ when in movement, then the movement model can estimate the travelled distance using $\Delta s_{MT} = v_{max} \cdot \Delta t$, where Δt denotes the time since the last invocation of the movement model. This is just an approximation of the travelled *distance*, as the movement detector unfortunately does not indicate direction. This implies that the action a_T introduced in Sect. 2 does not describe the directed movement between $t = T - 1$ and $t = T$, from position $l \in L$ to position $l' \in L$. Rather the implemented action model defines a_T as movement from one position $l \in L$ to all positions $l' \in L$ within a distance of s_{MT} .

As the movement sensor is not accurate concerning the travelled distance of the MT, the underlying uncertainty is modelled as two-dimensional Gaussian distribution, with mean $\mu = \Delta s_{MT}$ and $\sigma = 1$. The effect of applying the movement model is clarified in Fig. 2. The left illustration shows the belief about the MT's position at time $t = T - 1$. The probability map has two peaks, indicating that there are two positions the MT is likely to be located at. To reduce computational complexity the action model is only applied to probabilities above a certain threshold. In this example the action model is only applied to the maximum of each peak. The right illustration shows the belief at time $t = T$ after movement has been detected. The action model is applied to the two peak values and the resulting probability distributions are additively superimposed. The superimposition of the two volcano cones is apparent – note how the cones are weighted by the probability of their origin.

The outlined scheme provides a means to calculate the desired $P(l | a_{T-1}, L_{T-1} = l')$; tests have proven the estimation of the distance travelled by the MT's between two invocations to be sufficiently accurate.

Table 1. Object classes and their parameters.

object class k	description	attenuation η_k^{Wall}	c_k
1	window	6.0 dBm	2
2	thick concrete	15.0 dBm	7
3	doors	2.0 dBm	3
4	light wall	4.0 dBm	7
5	steel locker	7.0 dBm	1
6	thin concrete	5.0 dBm	1

4.3 Probability Map Initialization

Before the first iteration of the algorithm the probability map $Bel(L_{t=0})$ needs to be initialized to reflect the system’s knowledge (or at least belief) about the MT’s location. In principle two cases need to be considered. In the first case information about the MT’s location is available and can be used to initialize the map. This information can be provided by external intervention (e.g. querying the user about his location) or by means of a user profile. In the second case no information pertaining to the MT’s initial location is available, which is modelled by initializing $Bel(L_{t=0})$ equally distributed, i.e. all $l \in L$ are equally probable.

The current implementation has no notion of users but only locates terminals. This is done mainly for reasons of privacy, as the WhereMops-system is being used as a research and production platform. Hence the system cannot work on the basis of user profiles and initializes the probability map equally distributed.

5 Experimental Evaluation

This section presents the results of the experimental evaluation and compares them with the results obtained of related systems. The investigated properties include the absolute positioning errors both in case of line-of-sight (LOS) and without line-of-sight (NLOS) with no movement, the influence of the uncertainty factor in the radio maps and the accuracy of the estimated positioning error.

All experiments were conducted in the office wing of the Chair for Communication and Distributed Systems at Aachen University depicted in Fig. 1. The wing has a dimension of about $40\text{ m} \times 15\text{ m}$ and is covered by three access points placed on this floor, which are marked with circles in Fig. 1. The radio map was generated using the parameters listed in Table 1. The chosen cell size for all maps was $0.5\text{ m} \times 0.5\text{ m}$. Furthermore $n_P(d_0)$ was set to -37 dBm with $d_0 = 2\text{ m}$ and $\eta_{FS} = 2$.

The following describes the conducted experiments and their results. The positioning error is defined as the Euclidian distance between the true position and the absolute maximum of the probability density function describing the belief about the MT’s state.

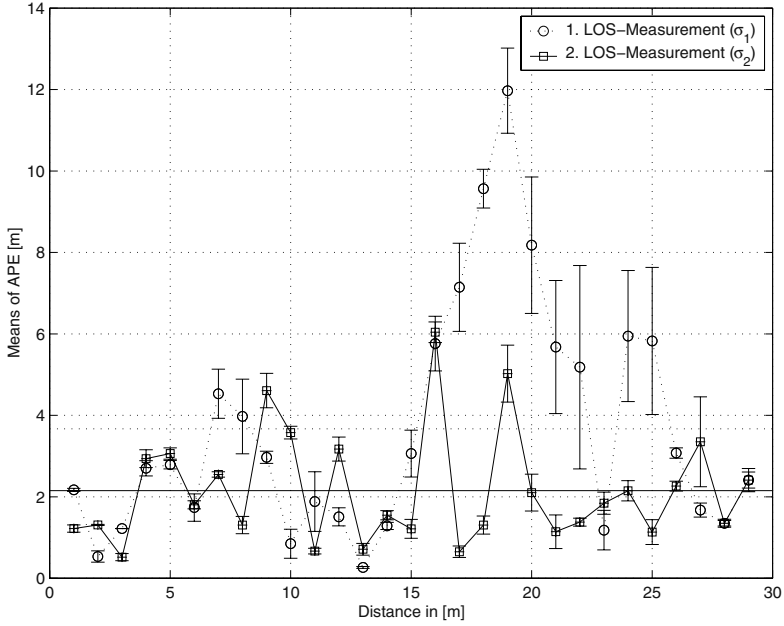


Fig. 3. Absolute positioning error under LOS conditions.

5.1 Results of LOS-Measurements

The LOS-measurements were conducted twice to examine the effect of the different uncertainty models (Eq. 11) and (Eq. 12) on the absolute positioning error. Though this series is categorized as being under LOS-conditions it must be noted that all times there is only line-of-sight to one access point.

The measurements were taken at locations in the hallway spaced apart by one meter and marked by a cross in Fig. 1. In both series of measurements the terminal was located periodically every 5 sec. After 10 iterations the current probability density, i.e. the belief about the terminal location, was reset by an equal distribution, to capture the convergence of the belief, i.e. probability distribution. In the first series this is done five times resulting in 50 location calculations per measurement point, while all other measurements repeated this procedure 10 times yielding 100 calculations per measurement point.

Figure 3 shows the mean of the 50 resp. 100 location calculations per measurement point. The error bars give the 95% confidence interval. The distance is given in relation to the LOS access point on the left side of the hallway outside the seminar room abbreviated by “SR”.

The first series of measurements started very promising with distance errors in the range of 1 m – 4 m, but then exposed great errors of up to 12 m in an area about 16 m – 22 m from the measurement’s origin. The estimated locations strongly tended towards the left-most access point, which indicates that the measured RSS is higher than the calculated RSS. Indeed the RSS readings of

the LOS access point were about 5–10 dBm higher than predicted by the model. As all other values in the radio map corresponded well with measured values, it is assumed that the deviation is caused by constructive multi-path propagation.

The experience gained from this first series of measurements motivated the change of the uncertainty model in the radio map module as described in Sect. 3.2. As the RSS measurements deviated more strongly from the empirical propagation model than estimated by σ_1 , the uncertainty model σ_2 was defined to assign higher uncertainties even to small transmitter-receiver separations. In the second series of LOS-measurements (using σ_2) the resulting error distance dropped significantly, especially in the previously problematic area. Though the effect of an incorrect estimation of the RSS is still visible, all remaining measurements were conducted using the new radio maps created with $\sigma_2(d)$.

5.2 Results of NLOS-Measurements

The case of NLOS is generally more challenging as the radio signals are subject to more transmission phenomena than in the case of LOS. Furthermore there is a higher chance of errors introduced by false parametrization of the radio propagation model. Despite these adverse conditions the results are quite satisfactory.

One series of measurement per room was carried out, where possible with the terminal placed in the middle of the room. At each location 100 measurements were conducted with a re-initialization of the probability map after 10 iterations (see above).

The cumulative histogram of the absolute positioning errors for the case of LOS and NLOS is shown in Fig. 4. The second series of LOS-measurements using the improved radio propagation model has a median error of 1.5 m and a 95th percentile of 6 m. The median error in the NLOS-case is 2 m whereas the 95th-percentile amounts to 8.5 m.

5.3 Estimated Positioning Error

A novel feature of the proposed indoor geolocation system is the ability to provide an estimate for the distance error of the calculated position. As stated above the maximum of the probability distribution was used as an estimate for the MT's location during system evaluation. To provide an estimate for the distance error, the probabilities of the grid cells surrounding the maximum were added up until a predefined probability threshold was reached. The result is a quasi-circular location area. The radius of this location area is interpreted as the estimated positioning error (EPE) with a confidence level corresponding to the threshold.

During the measurements the EPE at a 90% confidence level was recorded and afterwards compared to the absolute (i.e. true) positioning error (APE). If the EPE is lower than the APE, then the true position is outside the given location area. If the EPE is greater than the APE then the true position lies within the range stated by the algorithm. For the LOS-measurements 73% of the EPEs were greater than the corresponding APEs. This is quite satisfactory

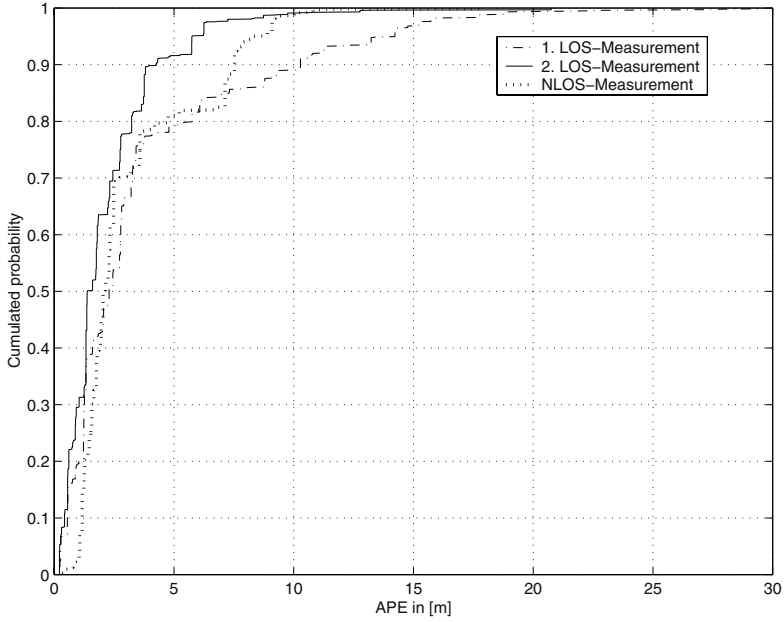


Fig. 4. Cumulative histograms of positioning errors under conditions of LOS and NLOS.

however it is below the desired confidence level of 90%. Surprisingly the NLOS-measurements showed better results in this respect. Here 89% of the EPEs were greater than the APEs.

5.4 Comparison

Table 2 compares the obtained results with those of related location determination systems presented by Bahl and Padmanabhan in [2] and by Youssef *et al.* in [4]. The location determination system proposed in this paper performs considerably better than the RADAR-system even in the case of NLOS. In contrast the schemes presented in [4] apparently outperform the Markov localization scheme.

Table 2. Comparison of positioning errors.

System	25 %	50 %	75 %	95 %
Markov-Localization LOS	0.6 m	1.5 m	2.8 m	6.0 m
Markov-Localization NLOS	1.5 m	2.0 m	3.6 m	8.5 m
RADAR-System	1.9 m	2.9 m	4.7 m	-
Joint Clustering	0.6 m	1.1 m	1.2 m	2.1 m
Incremental Triangulation	0.8 m	1.2 m	1.5 m	3.4 m

However it is very likely that this is due to the use of reference measurements for creating a radio map. It is questionable whether the high effort for creating radio maps by measurements justifies the achieved accuracy. Aachen University for example is spread over 150 buildings with a total usable floor space of around 300.000 m² – taking 300 sec reference measurements for every 2 m² as suggested in [4] would require at least 4.5 person-years assuming an eight-hour working day.

Furthermore the search-space of the implemented system is not as restricted as those in [2] and [4]. At all times every grid square presented a potential location of the mobile terminal. In contrast the works presented in [2] and [4] naturally restrict their search-space to those locations that were part of their offline measurements, which in both cases only covered hallways.

6 Conclusions and Future Work

Determining the position of mobile terminals within buildings on the basis of WLAN signal strength is extremely difficult, due to dynamic environments and complex radio propagation mechanisms. This paper presented the design and implementation of a probabilistic location determination algorithm based on the concept of Markov localization. To avoid time-expensive reference measurements a probabilistic radio propagation model was developed, which provides an estimate for the inaccuracies of the model.

The experimental results are very satisfactory especially considering the use of a radio propagation model instead of reference measurements to provide a data basis. The conducted experiments show that the proposed technique can provide a median error of less than 2 m even when there is no line-of-sight to an access point. The 90th-percentile is around 4 m under line-of-sight conditions.

In order to investigate the effects of the many parameters of the Markov localization algorithm more efficiently a simulator is currently being developed. The simulator will chose a position on a map and generate RSS values according to previously recorded histograms of real measurements. This simulator can also be used to evaluate the position estimation of a moving terminal, which can hardly be done in practice.

Finally the interpretation of the probability density representing the belief about the mobile terminal's position needs to be analyzed. The maximum of the density function is in general a good candidate for the location estimate. However finding the smallest location area with a given error probability could improve the accuracy.

References

1. IEEE 802.11 WG, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 8802-11: 1999 (ISO/IEC) (IEEE Std 802.11, 1999 Edition) Information technology - Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific Requirements – Part 11, 1999.

2. P. Bahl, V.N. Padmanabhan, *RADAR: An In-Building RF-based User Location and Tracking System*, Proceedings of IEEE Infocom 2000, Tel-Aviv, Israel.
3. A. Kishan, M. Michael, S. Rihan, R. Biswas, *Halibut: An Infrastructure for Wireless LAN Location-Based Services*, Technical Report, Stanford University, June 2001.
4. M.A. Youssef, A. Agrawal, A.A. Shanka, S.H. Noh, *A Probabilistic Clustering-Based Indoor Location Determination System*, Technical Report, CS-TR-4340, March, 2002.
5. P. Castro, P. Chiu, T. Kremenek, R. Muntz, *A Probabilistic Location Service for Wireless Network Environments*, Ubiquitous Computing 2001, Atlanta, GA, September 2001.
6. D. Fox, W. Burgard, S. Thrun, *Markov Localization for Mobile Robots in Dynamic Environments*, Journal of Artificial Intelligence Research 11, Pages 391–427, 1999.
7. M. Hassan-Ali, *Using Ray-Tracing Techniques in Site-Specific Statistical Modeling of Indoor Radio Channels*, Ph.D. Dissertation, Worcester Polytechnic Institute, Worcester, MA, 1998.
8. P. Harley, *Short Distance Attenuation Measurements at 900MHz and 1.8GHz Using Low Antenna Heights for Microcells*, IEEE JSAC, Vol. 7, No. 1, January 1989.
9. S.Y. Seidel, T.S. Rappaport, *914 MHz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings*, IEEE Transactions on Antennas and Propagation, Vol. 40, No. 2, February 1992.
10. K.W. Cheung, J.H.M. Sau, R.D. Murch, *A New Empirical Model for Indoor Propagation Prediction*, IEEE Transactions on Vehicular Technologies, September 1997.
11. M. Wallbaum, *WhereMoPS: An Indoor Geolocation System*, The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Lisboa, Portugal, September 2002.

A Topology Based Localization in Ad Hoc Mobile Sensor Networks

Sridhar Gangadharpalli, Uday Golwelkar, and Sridhar Varadarajan

Applied Research Group, Satyam Computer Services Ltd.,
14 Langford Avenue, Lalbagh Road, Bangalore 560 025 INDIA
{sridhar_gangadharpalli, sridhar, uday_golwelkar }@satyam.com

Abstract. Localization in an ad hoc mobile sensor network is an important requirement as most of the applications that use sensor data require sensor location information to complete the processing. A typical sensor network has over hundred to thousand sensor nodes, and considering the size and cost of a sensor, using only GPS for localization is not very attractive. The mobility of sensor nodes could lead to network topologies wherein accurate computation of absolute position of all the sensor nodes may not be possible. In this paper, we propose a topology based localization approach that suggests a best possible approximate position for sensor nodes for which computation of exact absolute position is not possible. We have identified four basic topological configurations that help compute position with varied degree of accuracy. These atomic configurations have been identified keeping in mind the simplicity of the computational procedures associated with these configurations. In order to put less demand on a computational capability of a sensor node, we suggest that only a pre-defined number of sensor nodes are compute-enabled (c-nodes) in the sense that they have adequate computational power. Similarly, only a pre-defined number of sensor nodes are GPS-enabled. In such a sensor network, the distributed computation of localization is achieved by distributing the computational requirements of individual sensor nodes across the c-nodes. Each sensor node strives to improve its localization by constantly monitoring its neighborhood and requesting an associated c-node to recompute position whenever neighborhood topology changes. We provide some initial results that bring out the merits of the proposed approach.

1 Introduction

Recent technological advancements and availability of wireless devices have increased the demand for self-organizing networks without the need for any dedicated infrastructure. These ad hoc networks consists of multiple nodes, with each node self-sufficient in terms of communication and computation powers, interact with each other in a cooperative way to address issues at network level and in interacting with a central station. A sensor network with wireless capability is a constrained wireless ad hoc network with limited power, communication, and computational abilities. Further,

the sensor network tends to be large in size with hundreds and thousands of sensor nodes. An ad hoc sensors network is static if the nodes that are part of the network, after an initial configuration, do not change their position. On the other hand, in the case of an ad hoc mobile network, the nodes of the network move arbitrarily, independent of each other, resulting in dynamic and ad hoc changes in the network topology. It is a challenging task to define a self-organizing network in the case of a mobile ad hoc network.

A smart sensor network [11] consists of a number of sensors spread across a geographical region and each sensor node has adequate intelligence. Such sensor networks are deployed in a variety of application scenarios such as (a) military sensor networks to detect the presence of hazardous material; (b) environmental sensor networks to detect and monitor environmental changes; (c) traffic sensor networks to monitor vehicle traffic on highways; and (d) surveillance sensor networks to monitor for security purposes. In each of these cases, it is required for a smart sensor node to communicate the sensed data (such as temperature and atmospheric pressure) to a central station for intelligent processing of aggregated data from multiple sensors and decision making. An important attribute of the data communicated by a sensor node is its current location that is essential in many of the applications such as traffic monitoring or surveillance.

GPS [4] is well-known approach for obtaining an absolute position of a node. When a receiver is outside a constellation of transmitters, standard iterative techniques may not converge to a correct solution. In this case it is required to use the known solutions to pseudo-range equations. [10] describes a five dimensional optimization procedure derived from the pseudo-range equations to compute position in the absence of navigation data. However, from the point of view of deploying this technology in a large sensor network where each sensor node is constrained by power, size, form factor, and cost factor, one can only selectively deploy this technology in a sensor network.

Localization refers to the problem of computing the position of a sensor node in an ad hoc network. In a large sensor network, it is not possible to configure the position of a sensor node even in the case of a fixed ad hoc network as the process of installation of sensor nodes might be just randomly dropping of the sensors over a region of interest. The position of a sensor node can be either an absolute position or a relative position. However, it is useful to determine the absolute position as relative position would have to be redetermined if there is a topology change even though the node under consideration might not have moved. Being able to know their absolute position is one of the important characteristics of the nodes of a self-organizing network.

Localization approaches depend on some sort of communication between anchor points (or reference points) and the node whose location needs to be determined. There has been significant research in studying location identification and some of these results are briefly described in the following. Bulusu et al [2] describe an approach wherein multiple nodes in a network that form a mesh serve as reference points and transmit periodic beacon signals containing their reference positions. From the beacon signals received by a node from a set of reference points, the node localizes to the region that coincides with the intersection of the connectivity regions of the set of reference points. Doherty et al [3] describe a method for estimating unknown node positions in a sensor network based on connectivity-induced constraints. Feasible solutions to the position estimation are determined using convex

optimization. [6] describes a self-localization method based on time-of-arrival and direction-of-arrival measurements by a subset of sensor nodes from a number of source signals placed at unknown locations. The method is for solving self-calibration problem with minimum number of sensor nodes and sources and provides an initial estimates for an iterative descent computation needed to obtain maximum likelihood calibration parameter estimates. [7] describes Ad hoc Positioning System that is a distributed, hop by hop positioning algorithm and works as an extension of both distance vector routing and GPS positioning in order to provide approximate location for all nodes in a network where only a limited fraction of nodes have self-location capability. [5] describes an algorithm in the context of a distributed sensor networks using which each sensor node determines its position in physical space based on their location in the network topology. The algorithm is based on determining, for each sensor node, the number of hops it is away from each of the basis nodes (those nodes that are aware of their location) and converting these hop-based distances into Cartesian coordinates. [9] describes a distributed technique for achieving fine-grain location awareness based on a limited fraction of beacons. The technique called as ad hoc localization system enables nodes to dynamically discover their own locations through ranging and estimation processes.

[8] describes a distributed algorithm for determining the positions of nodes in an ad-hoc, wireless sensor network in two phases: the startup-phase addresses the issues related to sparse availability of GPS-enabled nodes and uses a cooperative mechanism to spread location information of the anchor nodes throughout the network; and refinement phase addresses the issues related to reducing the error in initial position estimates.

[1] describes efficient algebraic tools for solving explicitly nonlinear geodetic problems such as GPS pseudo-ranging based on the algebraic techniques of Grobner bases and Multipolynomial resultants. The problems of localization can also be posed as a solution of a system of quadratic equations and the approaches suggested in [1] can be used to solve these pseudo-range equations.

In this paper, we propose a topology based localization approach. Briefly, we consider a sensor network in a field, with each sensor node having the property of wireless communication and low mobility. Some of the sensor nodes are GPS-enabled and are called as *g-nodes*. Similarly, some of the sensors are compute-enabled and these nodes are called *c-nodes*. Given this scenario, our objective is to achieve self-localization in each of the sensor nodes. Figure 1 depicts a typical sensor network. An *ng-node* is either a *c-node* or *w-node*.

2 Atomic Configurations

The process of localization is to be able to assign the absolute position to a node in a sensor network. Such a position is necessary for an intelligent sensor node to undertake location-specific sensing. Self-localization refers to a process wherein a node is able to establish its position based on the neighborhood information. In order to achieve self-localization, we propose to define a few topology based configurations

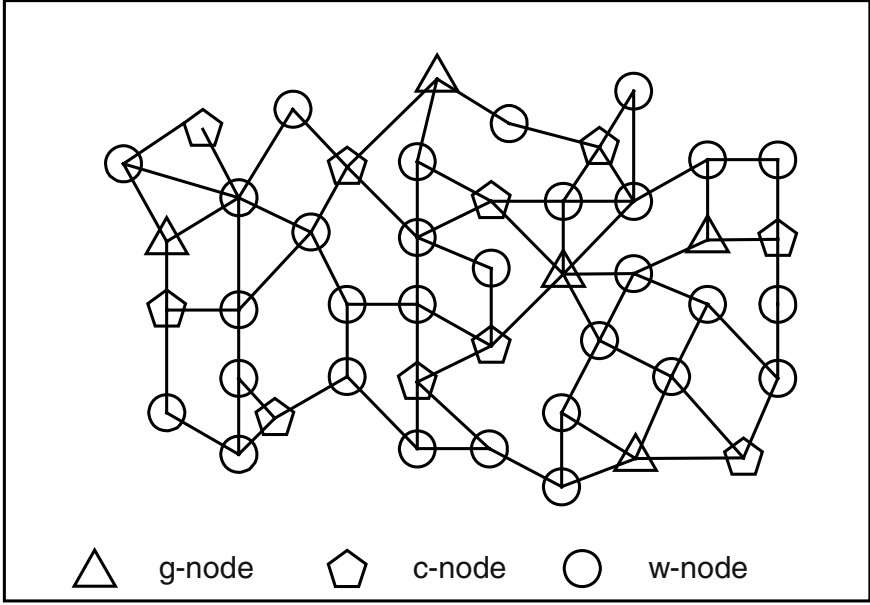


Fig. 1. Typical Sensor network. An ng-node is either a c-node or w-node

and embed a configuration identification procedure in each of the nodes. These configurations are called as atomic configurations as they are, in some sense, minimum amount of information that is required to localize a node within some bounds. The reason for identifying multiple atomic configurations is that in an ad hoc mobile sensor networks, depending on the various factors and the intended application, not all sensor nodes may be equipped with GPS capability. Addition of such a GPS feature would make a sensor node not only costly but also bulky and stresses the battery power. On account of the fact that only a few nodes are GPS-enabled and with sensor node mobility, there are certain topological possibilities in which exact computation of position of all the sensor nodes is not possible. In order to address such a situation, additional configurations have been identified so that where exact position identification is not possible, an approximate position could be assigned. An effort has been made to suggest atomic configurations keeping in mind computational simplicity. While position recomputation could be on rare occasions in a fixed sensor network, it is required to recompute the position quite often in the case of a mobile sensor network and this recomputation frequency not only depends on the self-mobility of a sensor node but also the mobility of the other nodes in the sensor network.

In the following, we describe four atomic configurations giving details such as configuration identification mechanism and position computation procedures. In the following, note that a g-node has been used to stand for a GPS-enabled node or location-aware node.

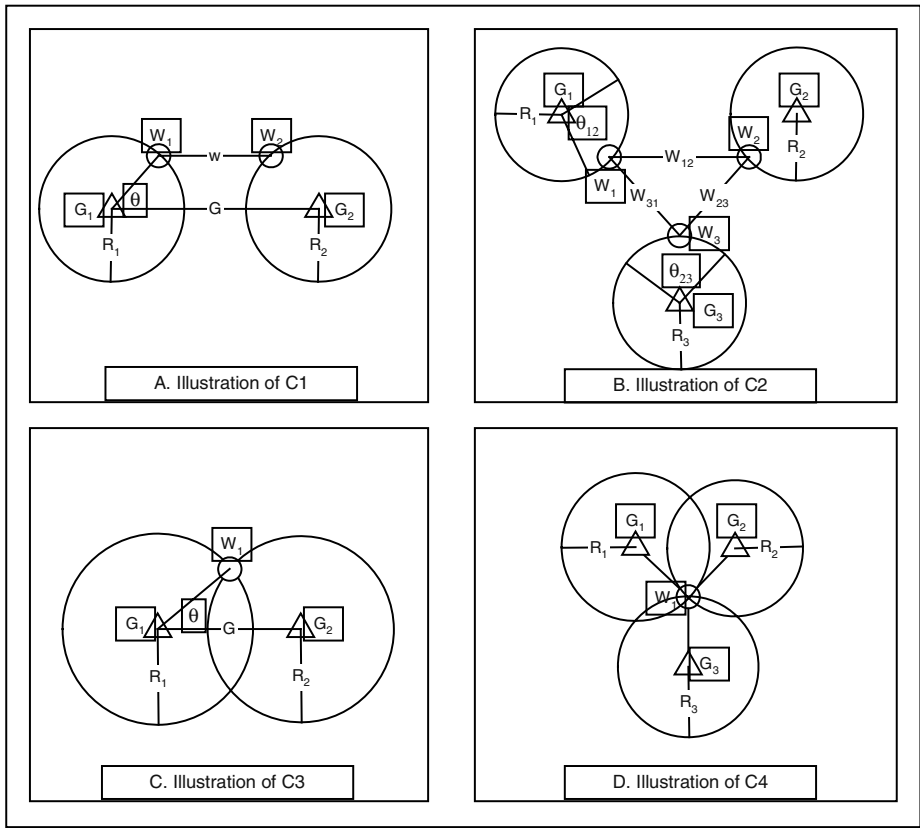


Fig. 2. Illustration of atomic configurations. A,B,C and D respectively illustrate the atomic configurations C1, C2, C3, and C4.

Configuration 1 (C1): A pictorial depiction of C1 is shown in Figure 2. Note that, in this configuration, there are two g-nodes and two w-nodes and W_1 is only within the range of G_1 and W_2 while W_2 is only within the range of G_2 and W_1 .

Configuration 2 (C2): A pictorial depiction of C2 is shown in Figure 2. Note that this configuration is an improvement over C1 in terms of the accuracy of position computation. In this case, there are three g-nodes (G_1 , G_2 , and G_3) and three w-nodes (W_1 , W_2 , and W_3) with the restriction that W_1 is only within the range of G_1 , W_2 , and W_3 and similar restrictions apply for W_2 and W_3 .

Configuration 3 (C3): A pictorial depiction of C3 is shown in Figure 2. Note that this configuration is an improvement over C1 and C2 in terms of the accuracy of position computation. In this case, a single w-node (W_1) is within the range of two g-nodes G_1 and G_2 thereby improving the position computation accuracy.

Configuration 4 (C4): A pictorial depiction of C4 is shown in Figure 2. Note that this configuration is an improvement over C1 through C3 in terms of the accuracy of position computation. In this case, one w-node (W_1) is within the range of three g-nodes (G_1 , G_2 , and G_3) resulting in a well-know configuration.

Configuration Identification

// Executed by w-node

// NeighborhoodData contains all possible instances of one or more identified configurations

{ Identify all the neighboring nodes (n-list) (1 hop away) and compute range from itself (w-node) to each of the identified neighboring node;

// each neighboring node is within the communication range of w-node

Let GN be the number of g-nodes in n-list;

If $GN = 0$ { return null set;}If $GN \geq 3$ { select three g-nodes from n-list;

Form NeighborhoodData with configuration id as C4; return NeighborhoodData; }

If $GN = 2$ { select two g-nodes from n-list;

Form NeighborhoodData with configuration id as C3; return NeighborhoodData; }

Let WN be the number of w-nodes in n-list; Let w-list be n-list without g-node;

While (w-list is not empty)

{ Identify two w-nodes in w-list such that they are within the range of each other and each of the two w-nodes has only one g-node within its range and let w2 be the list of these two w-nodes;

If w2 is not empty { Append w2 with configuration id as C2 to NeighborhoodData;

Remove w2 from w-list; Continue; }

Identify one w-node in w-list such that it has only one g-node within its range and let w1 be the list of this w-node;

If w1 is not empty { Append w1 with configuration id as C1 to NeighborhoodData;

Remove w1 from w-list; Continue; }

Return NeighborhoodData;

}

}

Fig. 3. Describes the steps involved in identifying the configurations in a sensor network. Note that this algorithm is executed by each w-node and hence, the configuration identification is from the point of view of a w-node.

Figure 3 describes the procedure for identifying the atomic configuration of the sub network.

In the following, we describe a computational procedure for each of the identified four atomic configurations.

In Algorithm C1, θ is computed by applying cosine rule to the triangle formed by W_1 with G_1 and G_2 as shown in Fig. 2. Observe that in the algorithms C1 through C3, the position of a w-node is not determined uniquely and hence, each of these algorithms provide a narrowed search space for computing the possible positions. Further, the multiple instances of the configurations C1 and C2 help in progressively reducing the solution search space. Also, multi-hop neighbors of a w-node whose locations are known would help in providing more constraints for reducing the search space.

<p>Computational Procedure for C1</p> <p>// Executed by c-node on behalf of a w-node</p> <p>{ Let R_1, G, W, and q be as shown in Fig. 2;</p> <p> Compute θ as</p> $\theta = \cos^{-1}((G^2 + R_1^2 - W^2) / 2R_1G)$ <p> $\varphi = (360 - 2*\theta)$;</p> <p> Return φ;</p> <p>// φ is the narrowed search space</p> <p>}</p> <p style="text-align: center;">A: Algorithm C1</p>	<p>Computational Procedure for C2</p> <p>// Executed by c-node on behalf of a w-node</p> <p>{ Let θ_{12} and θ_{23} be as shown in Fig. 2;</p> <p> Compute φ_{12} using Algorithm C1 between W_1 and W_2;</p> <p> Compute φ_{23} using Algorithm C1 between W_3 and W_2;</p> <p> Identify m points on arc φ_{12} and n points on arc φ_{23};</p> <p> Let NSS be NULL;</p> <p> For each point p in m</p> <p> For each point q in n</p> <p> { Compute distance d between p and q;</p> <p> If d is close W_{13} Add p to NSS;</p> <p> }</p> <p> Find arc A defined by NSS</p> <p> Return angle subtended by A;</p> <p>}</p> <p style="text-align: center;">B: Algorithm C2</p>
<p>Computational Procedure for C3</p> <p>// Executed by c-node on behalf of a w-node</p> <p>{ Let R_1, R_2, G, and q be as shown in Fig. 2;</p> <p> Compute θ using R_1, R_2, and G;</p> <p> Return q;</p> <p>}</p> <p style="text-align: center;">C: Algorithm C3</p>	<p>Computational Procedure for C4</p> <p>// Executed by c-node on behalf of a w-node</p> <p>{ Let W_1, G_1, G_2, and G_3 be as shown in Fig. 2;</p> <p> Compute the coordinates of W_1 by triangulation using G_1, G_2, and G_3 information;</p> <p> Return the coordinates;</p> <p>}</p> <p style="text-align: center;">D: Algorithm C4</p>

Fig. 4. Computational procedure for each of the identified four atomic configurations.

3 Self-Calibration of Position

Given a sensor network as shown in Figure 1, we describe an approach in which a sensor node uses the state of the sensor nodes in its neighborhood to self-calibrate its estimated position. The sensor node uses Configuration Identification Mechanism discussed in the previous section to improve its estimate over time. As the nodes move around, the state in the neighborhood changes and the node uses this change to its advantage to improve its position awareness. On initialization, the node performs the algorithm described in Figure 5 to obtain an initial estimate of its position. Note that the accuracy of its estimate depends on the neighborhood topology and due to the ad hoc nature of the mobile sensor network, this topology changes dynamically. After the successful initialization, the node constantly monitors for (a) any change in the neighborhood topology; and (b) any change in the position estimate of its neighbors. In either of the cases, it recalibrates itself using the algorithm described in Figure 5. Observe that in the algorithm described in Fig. 5, the node interacts with a c-node to help compute its position. Fig. 6 describes the algorithm executed by a c-node in order to compute the position of a w-node on receiving the inputs from the w-node. Note that c-node receives multiple instances of a configuration related data and

computes the possible angular position of w-node. Finally, when it has received and processed all the data sets, the resulting angular position is converted to possible positions and is returned to w-node.

```

Localization - Client
// Executed by w-node on need basis; Interacts with c-node assigned to w-node;
// A c-node is assigned during initialization statically and subsequently gets reassigned during
dynamic assignment
{ Analyze neighbors to determine the best possible configuration; // Use algorithm described
in Fig. 3 and obtain NeighborhoodData
  For each NeighborData in NeighborhoodData
  { Send NeighborData to c-node; Wait for Ack;
  }
  Receive PositionData;
  If Exact Position { Store Exact Position; return; }
  Analyze Possible Positions based on past position and mobility information;
  Store the analysis result as Feasible Positions;
}

```

Fig. 5. Localization Algorithm – Client

Observe that in the algorithm described in Fig. 5, a w-node interacts with a particular c-node for computational purposes. There are three ways to assign a c-node to a w-node so that position computation can be distributed across the available c-nodes. In a static assignment, a w-node is configured to have an associated c-node. While it is easier to achieve load balancing by equally distributing the w-nodes to the available c-nodes during configuration, depending on the topology, this may have an excessive traffic across the network.

A dynamic assignment, on the other hand, tries to minimize the network overloading and hence, reducing the delay in communicating the results back to a w-node. However, in this case, additional computational effort is required to achieve load balancing. As a compromise, in quasi-dynamic assignment, instead of reassigning whenever there is a change in topology, the reassignment is made at regular, long intervals.

Observe that this periodicity could vary over time based on the nature of mobility of sensor nodes. A distributed dynamic assignment algorithm is described in Figures 7 and 8.

C-nodes implement the position computation algorithms described in Section 2 for the various atomic configurations. Further, as it receives multiple data related to a particular configuration from a w-node, the c-node computes position for each of these data related to the configuration to finally return a best approximate value if the configuration under consideration is not C4.

Our initial simulations involved defining networks over a region of 100 square units with 120 nodes each with one unit range. We have implemented the algorithms related to the four atomic configurations and the configuration identification procedure. The following figures, Figs. 9 and 10, describe some of the experimental results.

```

Localization - Server
// Executed by c-node in response to data received from a w-node
{
    Receive NeighborData from w-node; Analyze the data and check for configuration;
    If no more of neighbor data from w-node
    { Convert ThetaSet to Possible Positions; return Possible Positions to w-node }
    Switch {
        Case C4: Use algorithm C4 and compute Position; return Exact Position to w-node;
        Case C3: Use algorithm C3 and obtain Theta;
        Case C2: Use algorithm C2 and obtain Theta;
        Case C1: Use algorithm C1 and obtain Theta;
    }
    ThetaSet = ThetaSet  $\cup$  Theta;
    Send ACK to w-node;
}

```

Fig. 6. Localization Algorithm - Server

```

Dynamic Assignment - Client
// Executed by w-node
{ Discover all c-nodes in the network;
  Arrange c-nodes in non-decreasing order of hops into c-list;
  // c-node and w-node are separated by a number of hops;
  Start Timer; // set to available time before which all w-nodes should have their c-nodes assigned;
  // this time includes guard band time also
  While (not yet assigned a c-node)
  { Select next c-node from c-list; Send assign request to c-node;
    Wait for ACK or NACK
    If ACK is received {c-node gets assigned; return; }
    If Timer expires { Reassign previously assigned c-node; return; }
  }
  Reassign previously assigned c-node;
}

```

Fig. 7. Algorithm for dynamic assignment at client

Note that, in the Figs. 6A and 6B, the nodes connected by solid lines indicate that the position computation is by using C4 configuration, the nodes connected by long broken lines indicate that the position computation is by using C3 configuration, and the nodes connected by dotted lines indicate that the position computation is by using C2 and C1 configurations.

```

Dynamic Assignment - Server
// Executed by c-node; w-list contains granted requests
// ow-list contains pending requests in the received order with the most recent at the head
// W is the limit number of w-nodes that can be assigned to c-node

{ Start Timer; // set to available time before which all w-nodes should have their c-nodes assigned
  While (1)
  { While (1)
    { Check for request; If received Break;
      If Timer Expires Break both loops; Sleep (0);
    }
    if |ow-list| < W {Add request.w-node at the head of ow-list; Continue; }
    For each w-n in ow-list
    if w-n.Hops > request.w-node.Hops { Remove w-n from ow-list; Send NACK to w-n;
      Add request.w-node at the head of ow-list; Break;
    }
  }
  // Timer has expired
  If |ow-list| >= W { Copy W requests from ow-list to w-list on first-come-first-served basis;
    Send ACK to each of the w-nodes in w-list;
    Send NACK to the remaining requests in ow-list;
    Check for requests from w-nodes and for each such request send NACK;
    // these are the requests sent by w-nodes just before timer expiry to c-node
    Return; }
  // c-node can accommodate a few more w-node requests received within Guard Band time interval
  Start Timer // set to guard band time interval;
  While (1)
  { Check for request;
    If received request
      If |w-list| < W { Add request.w-node to w-list; send ACK to request.w-node;}
      Else Send NACK to the corresponding w-node;
    Else Sleep (0);
    If Timer Expires Break;
  }
}

```

Fig. 8. Algorithm for dynamic assignment at server

It is observed that, in Experiment 2, there were more number of nodes that were localized using C2 and C1 configurations as compared with those of Experiment 1.

4 Conclusions and Further Work

In this paper, we have proposed a topology based localization approach in the context of ad hoc mobile sensor networks. In a mobile sensor network in which there are a limited number of nodes with GPS-capability, there is a need for localization procedure that addresses the issues related to topologies in which exact computation

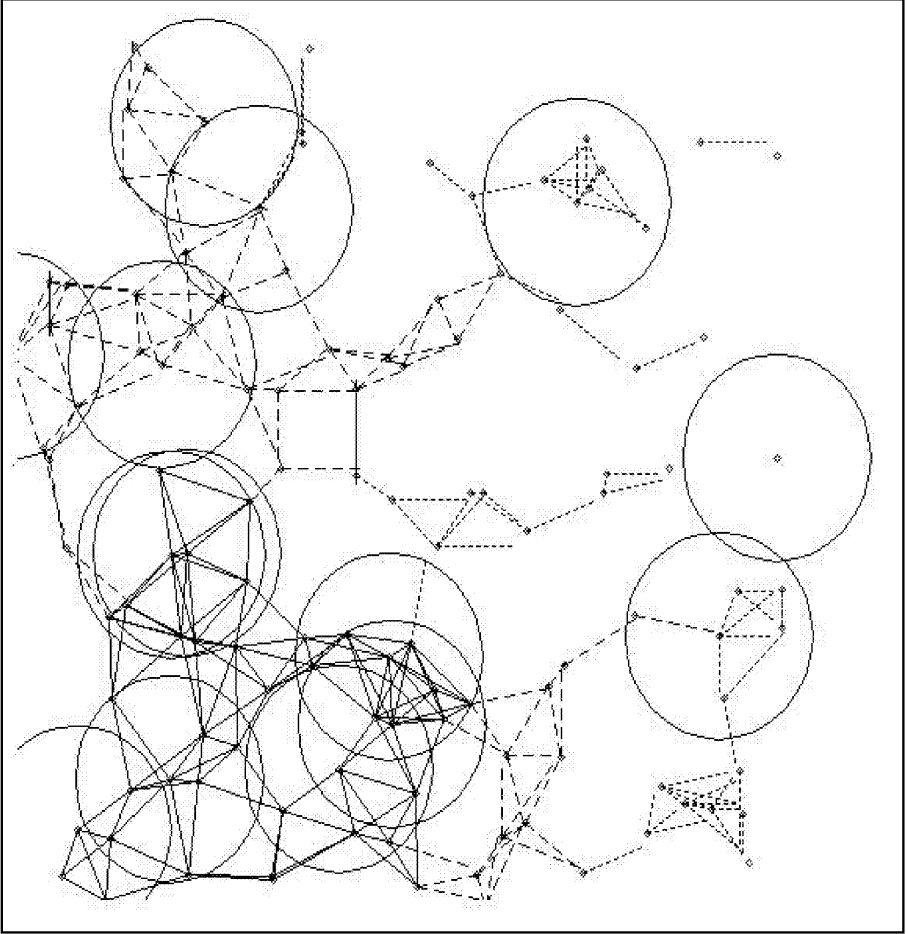


Fig. 9. Results of experiment 1

of position of many sensor nodes is not possible. Under these conditions, the approach suggested in this paper provides a best possible approximate estimate of the position of such sensor nodes. In order to achieve cost minimization while deploying a sensor network, we have suggested that the nodes in a sensor network could be of three types: g-nodes – those nodes that are GPS-enabled or location-aware; c-nodes – those nodes that have adequate computation power; and w-nodes – a wireless mobile sensor node.

We have suggested algorithms for atomic configurations that get executed on a c-node on behalf of a w-node during the self-computation of the position by the w-node. The idea behind identifying atomic configurations is to simplify the computational

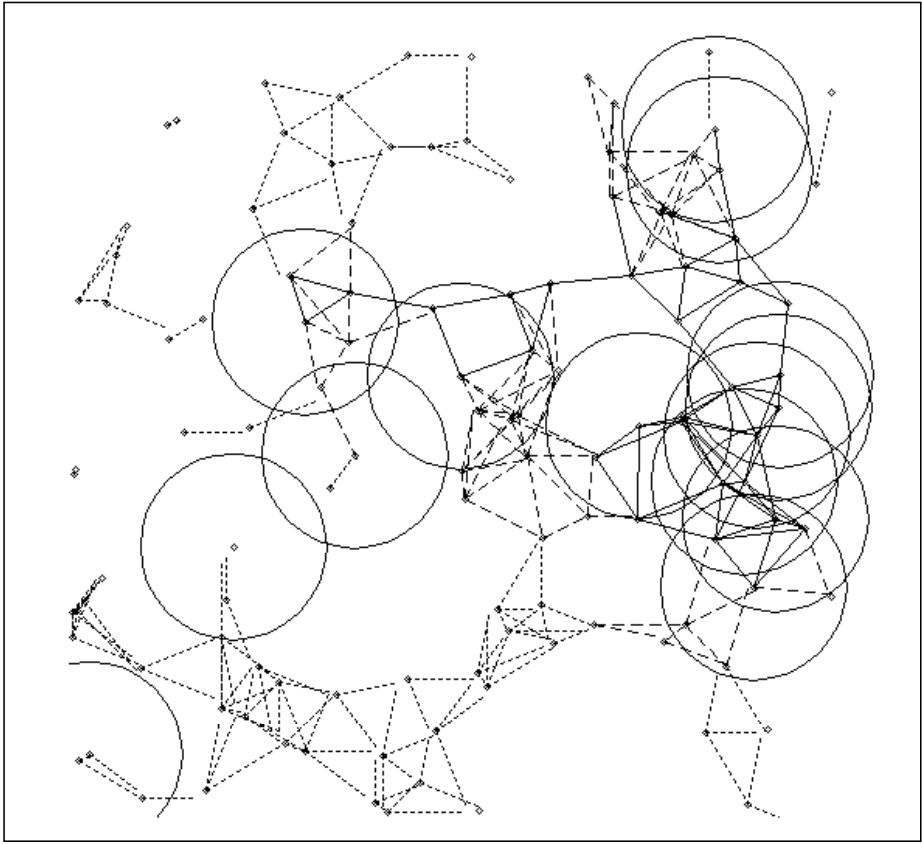


Fig. 10. Results of experiment 2

effort and handle the availability of only limited information. We have also suggested a dynamic assignment algorithm that dynamically, in a distributed fashion, assigns the best possible c-node for a w-node so that an attempt is made to achieve load balancing and minimize traffic across the network due to the interaction between w-nodes and c-nodes for position computational purposes. The results provided in this paper suggests that there are several w-nodes that require the usage of C1 and C2 configurations, along with other constraints, to determine their position. As part of the ongoing simulation work, we are working towards identifying different network topologies with a sufficiently large number of nodes and measuring the effectiveness of the suggested approach in terms of the extent of application of different configurations in localizing the nodes. Further, we are investigating the utility of distinct graph patterns such as (a) densely connected subnets; (b) sparsely connected subnets; and (c) partially localized chains. We also intend to explore the effects of density of w-nodes and sparsity of g-nodes on localization.

References

1. Awange, J.L., "Gröbner bases, multipolynomial resultants and the Gauss-Jacobi combinatorial algorithms – adjustment of nonlinear GPS/LPS observations," Ph. D. Thesis, University of Stuttgart, 2002.
2. Bulusu, N., J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.
3. Doherty, L., K. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," *Proceedings of IEEE INFOCOM 2001*, volume 3, pages 1655–1663, Anchorage, Alaska, April 22–26 2001.
4. Hofmann-Wellenhof, B., H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, Fourth Edition, Springer-Verlag, 1997.
5. McLurkin, J.D., "Algorithms for Distributed Sensor Networks," Masters Thesis for Electrical Engineering, University of California, Berkeley, December 1999.
6. Moses, R.L., D. Krishnamurthy, and R. Patterson, "A Self-Localization Method for Wireless Sensor Networks," *Eurasip Journal on Applied Signal Processing*, Special Issue on Sensor Networks, No. 4, Vol. 2003, March 2003.
7. Niculescu, D. and Badri Nath, "Ad hoc positioning system (APS)," *Proceedings of GLOBECOM*, San Antonio, November 2001.

Grcmob: A Group Mobility Pattern Generator to Evaluate Mobile Ad Hoc Networks Performance

Juan-Carlos Cano, Pietro Manzoni, and Miguel Sanchez

Department of Computer Engineering, Polytechnic University of Valencia
Camino de Vera s/n, 46071 Valencia, Spain
{jucano, pmanzoni, misan}@disca.upv.es

Abstract. We present an analysis of the behavior of a routing protocol when group mobility is involved. We concentrate on group mobility because there is a growing attention on the development and evaluation of MANET's approach applied to *personal area networks* (PANs), especially based on Bluetooth technology.

We propose four different group mobility models and describe a mobility pattern generator called **grcmob** to be used with the ns-2 simulator. We compare the behavior of a classical reactive routing protocol, the *Dynamic Source Routing* (DSR) protocol and we perform a thorough evaluation of its behavior using as a reference the behavior obtained with the random waypoint mobility model.

We observe the high variability of the results and the need to know exactly the behavior of the system and the impossibility to define a unique proposal which is general to whatever environment. We make evident that also the mix of inter- and intra-group communication has a strong impact on the routing protocol performance and should therefore be taken into consideration when tuning or designing a routing protocol. Finally, we demonstrate that the presence of groups forces the network topology to be more sparse and therefore the probability of network partitions and node disconnections grows.

1 Introduction

Mobile ad hoc networks (MANETs) are an example of mobile wireless networks that do not require any fixed infrastructure, which means that their topologies can vary randomly and at unpredictable times. The *Internet Engineering Task Force* (IETF) MANET working group [1] proposed various routing protocols for *ad hoc* networks during the past few years. The evaluation of most of these proposals has been performed with the aid of various network simulators. Most of these tools, such as the ns-2 [2] or the GloMoSim [3], make use of synthetic models for mobility and data patterns.

However, the general problem of modelling the behavior of the nodes belonging to a mobile network has not a unique and straightforward solution. Mobility

patterns depend on various factors like the physical environment, the user objectives, and the user inter-dependencies. Hong et al., [4] showed that these models can have a great effect upon the results of the simulation, and thus, on the evaluation of these protocols. In [5] a survey of the existing mobility models is presented.

The mobility models that are commonly used to simulate MANETs can be classified into two categories: individual-based and group-based. An individual-based model describes node mobility independently of any other nodes. With group-based mobility models, individual nodes movement is depended on the movement of close-by nodes.

The objective of this work is to show the impact of group mobility on the behavior of a routing protocol and to present the critical factors that must be taken into consideration when optimizing the behavior or in general the design of a routing protocol for MANETs. We compare the results with the classic *Random Waypoint* [6] model without groups to simply provide a reference to better understand the obtained results. We concentrate on group mobility because there is a growing attention on the development and evaluation of the MANETs approach applied to *personal area networks* (PANs), especially based on Bluetooth technology [7]; consider for example the work of Gerla et al., [8]. PANs exploit the concept of “piconets”, that is a very small-area network, normally of up to 8 nodes, where a dedicate node is the “master” inside the topology. Piconets can be joined together to form “scatternets”. This types of networks emphasize the group-behavior of the network and therefore reinforce the need for more dedicated mobility models.

We describe four different group mobility models: the *Random Waypoint Group Mobility Model* (RWG), the *Random Direction Group Mobility Model* (RDG), the *Manhattan Group Mobility Model* (MHG) and the *Sequential Group Mobility Model* (SQG). The RWG model extends the classic random waypoint model applying mobility to a subset of close-by nodes at a time. While with the RWG model a group destination is normally inside the movement area, with the RDG model we stretch the final destination to a border of the movement area. The MHG model forces movement to be only along vertical or horizontal directions. Finally, the SQG model applies the RWG approach to the groups in sequence, i.e., groups are ordered and group i has to move toward the current position of group $i - 1$.

We consider a classical reactive routing protocol, the *Dynamic Source Routing* (DSR) protocol, and we perform a thorough evaluation of its behavior under the four proposed group mobility models using as a reference the behavior obtained with the random waypoint model. We observe the high variability of the results and the need to know exactly the behavior of the system and the impossibility to define a unique proposal which is general to whatever environment. We make evident that group mobility pattern highly affects the performance of a routing protocol but also that the mix of inter- and intra-group communication has a strong impact on the routing protocol performance and should therefore be taken into consideration when tuning or designing a routing protocol. Finally, we

demonstrate that the presence of groups obviously forces the network topology to be more sparse and therefore the probability of network partitions grows. This phenomenon is especially evident with the SQG mobility model.

The rest of this paper is organized as follows: Section 2 describes the related work dedicated to the analysis of the impact of group mobility over MANETs. Section 3 describes the mobility models we propose, the software tool we designed and outlines the problems with group mobility. Section 4 presents the sensitivity analysis over the performance of DSR with our four mobility models and finally, Section 5 presents the conclusions of this work resuming a few considerations over the approach to be followed to optimize routing protocols.

2 Related Work

The most widely used individual-based mobility model is the *random waypoint* model where motion is characterized by two factors: the maximum speed and the pause time. Each node starts moving from its initial position to a random target position selected inside the simulation area. The node speed is uniformly distributed between 0 and the maximum speed. When a node reaches the target position, it waits for the pause time, then selects another random target location and moves again. Many other variations of this model exist which increase the randomness of the mobility process. For example the Random Direction model [9], the Smooth Random Mobility Model [10], or the Random Gauss-Markov Mobility [11].

In a previous work [12] we intuitively described several group-based models like the column model, the pursue model, and the nomadic models. In the first model the nodes form around a reference grid (in this case, a 1-d line), and roam within a constant distance of their point on the grid. When the grid moves, the nodes follow. In the pursue model, a particular node is moving according to one of the independent mobility models, and all other nodes are following this node. Their movement is described in terms of the vector of their current velocity, and the addition of an acceleration and random vector. In the nomadic model, which (along with the others listed above) is a less general form of the Reference Point Group Mobility Model [13], each node is associated with a logical base position. Each base position may have more than one node associated with it. The base positions themselves move according to some mobility model, and nodes associated with each base position stay within some predetermined distance of the base position, moving along with it.

Most of the research in ad-hoc networks is done by using individual mobility models because the simulation code is readily available and because group mobility adds even more parameters to take care of. To the best of our knowledge only two group mobility models have been described in the literature.

The first one, by M. Bergamo et al. [14], is called the *Exponential Correlated Random* (ECR) model and simulates the movement of nodes in a multihop packet radio network in a tactical setting. The model can have several groups of nodes. Each group as a whole moves according to the model, and each node

within a group also moves according to the model, but following the trajectory of the group. Given the current position of a node, the next one is calculated as:

$$b(t+1) = b(t) \cdot e^{-\frac{1}{\tau}} + (\sigma \sqrt{1 - e^{-\frac{2}{\tau}}}) \cdot r$$

where: b defines the position, τ the location-change rate, and r is a Gaussian distribution with variance σ . A pair (τ, σ) must be defined per each group. The main drawback of this model stands in the complexity to impose a given motion pattern by setting up the proper values for the model parameters.

The second group mobility model presented by X. Hong et al. [13], is denominated *Reference Point Group Mobility* (RPGM). This model presents a general framework for group mobility and can be used to simulated a wide range of mobility models. It defines the concept of group center (*reference point*), as a virtual point that moves following a set of *waypoints* (group motion). Group member experience random deviations from group motion. This model can be used in a variety of ways, as different scenarios can be represented (i.e., meeting room, exposition visit, isolated static groups, etc). The RPGM main drawback is that node motion within a group is restricted to relative low speed motion. Besides, this model leaves too many open parameters, so a lot of choices have to be done to completely specify a simulation setup. The RPGM model can generate topologies of ad-hoc networks with group-based node mobility for simulation purposes, it is not easy to use for partition prediction purposes [15]. This model supposes the presence of an omniscient observer, a so called *God*, which maintains the complete information about the mobility groups including their member nodes and their movements. Due to the distributed nature of these types of mobile networks, such high-level information is not easy to be made available to any mobile nodes at run-time. Moreover, the RPGM model represents the mobile nodes by their physical coordinates. Given only the instantaneous physical locations of the nodes, it is difficult to derive the characteristics the movement of the nodes' group movement.

3 The Group Mobility Models

In this work we present 4 different group mobility models which combine the random waypoint model with the concept of group. The models are:

1. The *Random Waypoint Group Mobility Model* (RWG): this model extends the classic random waypoint model applying mobility to a subset of close-by nodes at a time. This is the most straightforward extension which allows to make evident the characteristic of intra- and inter-group data-traffic.
2. The *Random Direction Group Mobility Model* (RDG): while with the RWG model a group destination is normally inside the movement area, with the RDG we stretch the final destination to a border of the movement area. This modification allows to stress routes extensions while reducing the “density waves” [16] effect.

3. The *Manhattan Group Mobility Model* (MHG): the MHG model forces movements to be only along vertical or horizontal directions. We are modelling a constrained environment where paths can follow only predetermined directions, like in downtown areas.
4. The *Sequential Group Mobility Model* (SQG): finally, the SQG model apply the RWG approach to all the groups in sequence, i.e., groups are ordered and group i has to move toward the current position of group $i - 1$. Figure 1 shows a sequence of three *nam* screen-shots which represent the evolution of the network topology when using the SQG model.

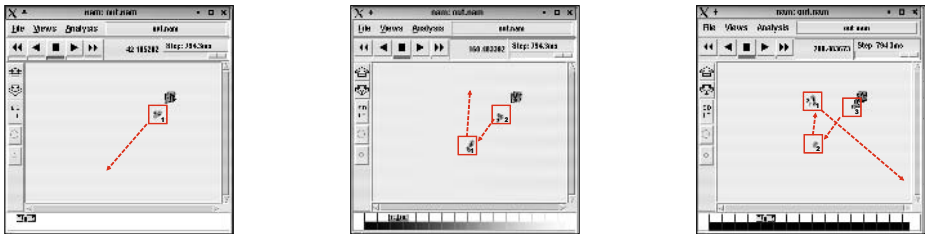


Fig. 1. Sequence of three *nam* screen-shots which represent the evolution of the network topology when using the SQG model.

We designed a mobility pattern generator, called **grcmob**¹, to be used with the ns-2 simulator whose approach is similar to that of the **setdest** module defined by CMU Monarch projects. The user has to define the number of groups, the total number of nodes, the simulation time, the area size, the max speed value and an initial position flag. We assume each group to have a fixed size, i.e., a fixed number of members; nodes are assigned evenly to each group. The initial position flag refers to whether we want to chose a random initial position for groups or we want the same initial position for every group. The concept of group, which can be informally described as a set of *close-by* nodes, is represented in **grcmob** using the notion of *sensitivity*. We introduce three parameters to characterize sensitivity: the *distance_group_sensitivity*, the *group_speed_sensitivity*, and the *group_init_motion_sensitivity*. First of all a single node is used as a reference for the other members of the group. The criteria to chose the reference node is irrelevant; in our case was the node with the lowest id. The *distance_group_sensitivity* indicates the maximum distance between the reference node and any other node in the group. The *group_speed_sensitivity* and the *group_init_motion_sensitivity* parameters are used to give flexibility to the relative movement of each of the member of the group. The first one expresses the range of values for each node speed with respect to the reference node, while the second one expresses when a node starts moving with respect to the reference node.

¹ The **grcmob** source code is available at <http://www.grc.upv.es/>.

The presence of groups raises an important issue related to the percentage of data traffic that is sent and received inside the same group, which we will call, *intra-group* data traffic, and the percentage of data traffic that is sent from one group and received inside a different group, which we will call, *inter-group* data traffic. The combination of these two types of traffic strongly impact on the routing protocol. The basic idea is that with intra-group data traffic no actual routing is required because the sender and the receiver are 1 hop away, while if we have a high percentage of inter-group data traffic, the number of hops will increase thus requiring more complex routing protocol. For this reason in the simulations we emphasized the evaluation of the average hops count.

4 Simulations

This Section reports the results of the sensitivity analysis we performed adopting the four mobility models described in Section 3 and using the DSR [6] routing protocol. We fixed to 100 the overall nodes number and employed 20 sources which generated 50% of intra-group data traffic and 50% of inter-group data traffic. We used 4 packets/seconds *Constant Bit Rate* (CBR) data flows with a packet size of 512 bytes. The source data traffic generating pattern was kept unchanged across all simulations.

The group sensitivity parameters were set to describe dense and stable groups. The *distance_group_sensitivity* was set to 50 meters, the *group_speed_sensitivity* was ± 0.15 meters/seconds and the *group_init_motion_sensitivity* was ± 0.15 seconds.

The overall mobility process, as for the random waypoint model, is based on alternating mobility periods and pause periods. The maximum duration for the pause periods, defined by parameter *pause_time*, was set to 20 seconds. This value was obtained by the work described in [16] to improve stability of the results. As a general rule we waited for each node of the group to have completed its movement phase before establishing the next movement for the whole group.

We defined a basic scenario (see Section 4.1), and modified one at a time the following parameters: the node speed, the number of groups, and the simulation area size. The objective was to determine how a specific single parameter affects the results. Regarding the performance metrics we concentrated on: the delivery rate, the route hops count and the end-to-end delay. The delivery rate is obtained by the ratio of the number of data packets delivered to the destination nodes divided by the number of data packets transmitted by the source nodes.

The simulation duration was set to 2000 seconds. During the first 1000 seconds the nodes only moved around and no data traffic was generated. According to [16] this would allow for the system to get to a stable state before data traffic is generated.

4.1 The Basic Scenario

In this section we describe the basic scenario which is used as a reference for the sensitivity analysis process. We supposed to have 20 groups over an area of

1000 meters \times 1000 meters and that nodes speed was equal to 3 meters/seconds. Figure 2 shows the results for each mobility model in terms of the data packet delivery ratio, the average hops count, and the average end-to-end delay.

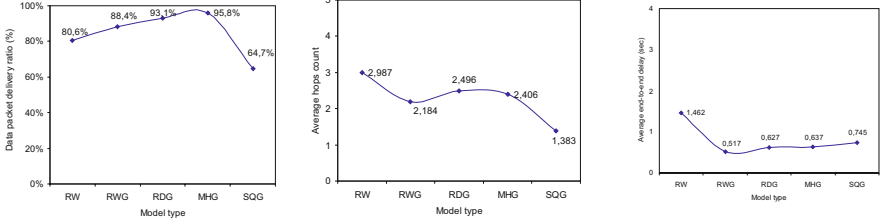


Fig. 2. Performance results for the 100 node MANET basic scenario using the different mobility models.

The random waypoint model shows the highest hops count. This is because all the data packets can potentially need several forwarding nodes. On the other hand, those scenarios using any of the group mobility model have a mixture of intra-group data packets (where no forwarding nodes are required) and inter-group data packets, thus the average hops count decreases with respect to the random waypoint case.

We can also observe that in general the end-to-end delay increases as the hops count increases. With the random waypoint model the delay can be almost three times higher with respect to group mobility models. This is mainly due to the fact that this mobility model suffers the effect of the “density waves” [16]. This phenomenon makes nodes to group around the center of the simulation area thus increasing the level of network congestion multiplying access interference.

In general we cannot observe any significant difference between the RWG, RDG and MHG models. This could be due to the relatively low number of groups that tend to make these scenarios similar. As we will select more dense scenarios we expect some differences to appear especially between the RWG model, where nodes tend to move toward the center of the area, and the RDG model, where nodes travels up to the border of the simulation area.

Finally, the SQG model presents the lowest delivery ratio and hops count. The end-to-end delay of the SQG comes from the high variability that exhibit intra- and inter-group data traffic. Most of the successfully delivered data packets are those from the intra-group connection. On the other hand, the low node speed of the basic scenario makes the SQG model quite sensitive to network partition, so a high percentage of the inter-group traffic do not succeed. Moreover, those inter-group packets that finally succeed have been waiting in intermediate queues for a

longer period of time, increasing thus the average end-to-end delay. It is expected that as node's speed increase network partition of this scenario decrease.

The above results must be analyzed taking into consideration the following points:

- with any of the four group mobility models, the 100 mobile nodes are distributed over 20 groups, thus making the resulting network topology much more sparse with respect to the network topology where the 100 mobile are not grouped.
- most importantly, the communication pattern has been selected randomly, with the only requirement of equally balance the inter- and intra-group communication. As stated before, we have 20 sources which generate half of the traffic inside the group and half of the traffic toward external nodes, thus 50% of the data packets do not need any forwarding node to be successfully delivered.

Varying the traffic distribution the performance results vary accordingly. Figure 3 shows the obtained results when varying the percentage of the inter- and intra-group traffic among values 0%, 25%, 50%, 75%, and 100%.

The traffic delivery rate drops below that of the random waypoint when the percentage on inter-groups traffic exceeds 60%. The presence of groups obviously forces the network topology to be more sparse and therefore the probability of network partitions grows. If we consider the average hops count, increasing the percentage on inter-groups traffic can lead the routing protocol, like in the case of the RWG, RDG, and MHG models, to perform worse than in the random waypoint case. A consequence of the increased value for the average hops count is the increment of the end-to-end delay.

4.2 Impact of Nodes Speed

In this section, we explore the effect of varying nodes speed over the basic scenario. Figure 4 shows the obtained results when varying the maximum node speed among 3 (basic scenario), 6, 9 and 12 meters/seconds.

Except for the SQG model, all the scenarios present a descendent trend for the delivery rate and the average hops count when node speed increases. This happens because as node speed increases, packets with longer routes could suffer from broken links with the possibility for packets to be dropped.

The four group mobility models behave better than the scenario where no group is selected. The reason mainly stands in the traffic distribution. The traffic model distributes the total traffic to be 50% intra-group and 50% inter-group. Thus, 50% of the packets do not need any forwarding node. It is also important to note that for those scenarios based on groups the data packet delivery ratio is not as high as one would expect because the 50% of the packets (inter-group data packets) could suffer from transient partition that exist in sparse networks.

As node speed increases, the RDG model increases the average hops count with respect to the RWG and the MHG. Nodes that follow the RDG model will

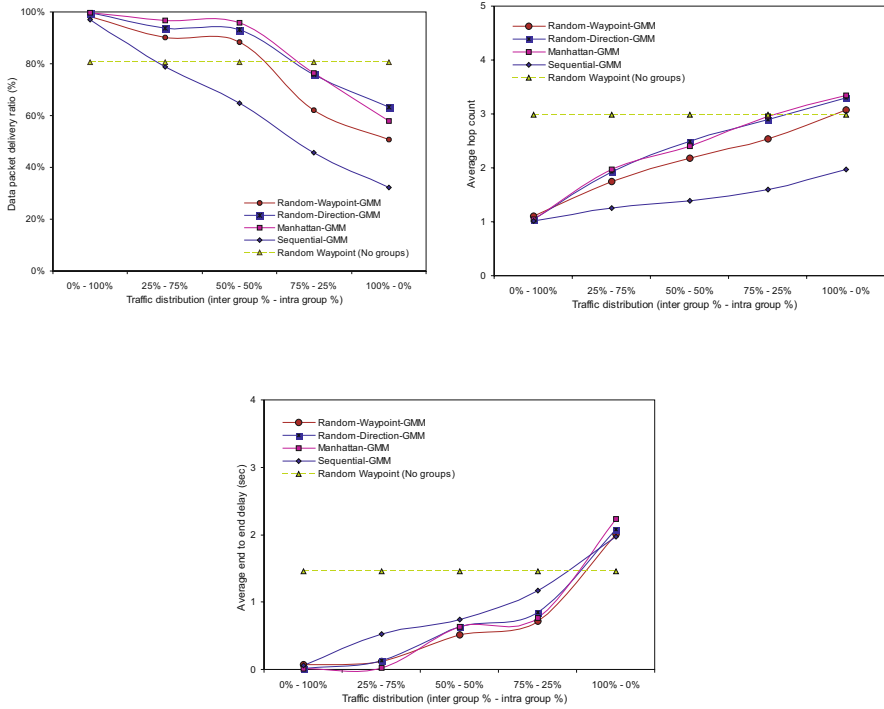


Fig. 3. Performance results for the 100 nodes MANET with five different mobility models as a function of traffic distribution

move up to the simulation area border thus increasing the average number of hops and so the end-to-end latency.

The SQG model behaves better as node speed increases in terms of delivery ratio and end-to-end delay. In this mobility model all the groups follow similar paths, thus as node speed increases, the distance among groups decreases, and the model tends to eliminate the partition that appear when the speed of nodes is low.

Finally, when looking at the details of the average end-to-end delay we can observe that all the models except the SQG increase the latency as node speed increases. As node speed increases, more packets have to wait in intermediate queues for the availability of new paths after a route breakage. However, the effect of congestions observed in the basic scenario using the random waypoint model tend to disappear at high speed because traffic tends to be more evenly distributed due to node's movement.

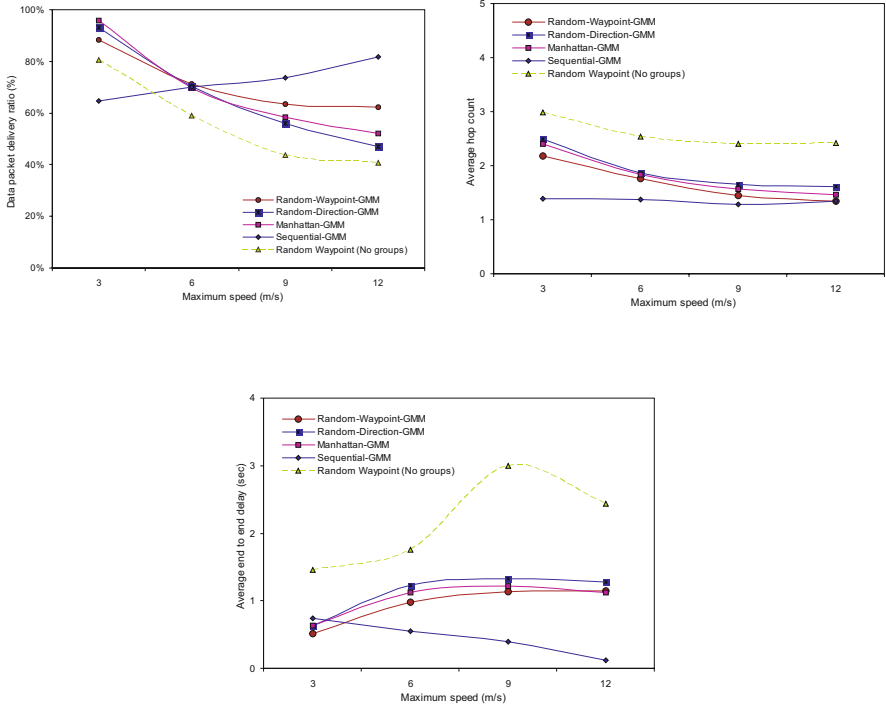


Fig. 4. Performance results for the 100 nodes MANET with five different mobility models as a function of the maximum node speed

4.3 Impact of Groups Number

We now evaluate how the number of groups can affect performance. Figure 5 shows the obtained results when varying the total number of groups among 1, 10, 20 (basic scenario), and 50. The performance results obtained with the random waypoint model will obviously not be affected. Similarly, when we select just 1 group, all the traffic become intra-group, independently of the mobility model. In that case, the average hop count is 1 hop and nearly 100% of the total packets can be successfully delivered.

As we increase the number of groups, the effect of transient partitions will decrease. As an example, the scenario where we select 50 groups the performance for the four group mobility models approach the random waypoint scenario. However Figure 5 shows that there are still differences. These differences are mainly due to the fact that still 50% of the total traffic do not need any forwarding node. So all the approaches based on groups get better performance in terms of delivery ratio, average hops count and average end-to-end delay.

We can also observe that in the 50 groups scenario, the RDG model gets worst performances in terms of hops count than all the other approaches. This is due to the fact that all nodes in this dense scenario move until they reach the border of the simulation area, thus increasing the average hops count and the end-to-end delay.

The scenarios where only 10 or 20 groups are selected, the RWG, RDG, MHG, and especially the SQG suffer from transient network partitions. This effect is even more visible at low speeds and will provoke packets to be periodically dropped.

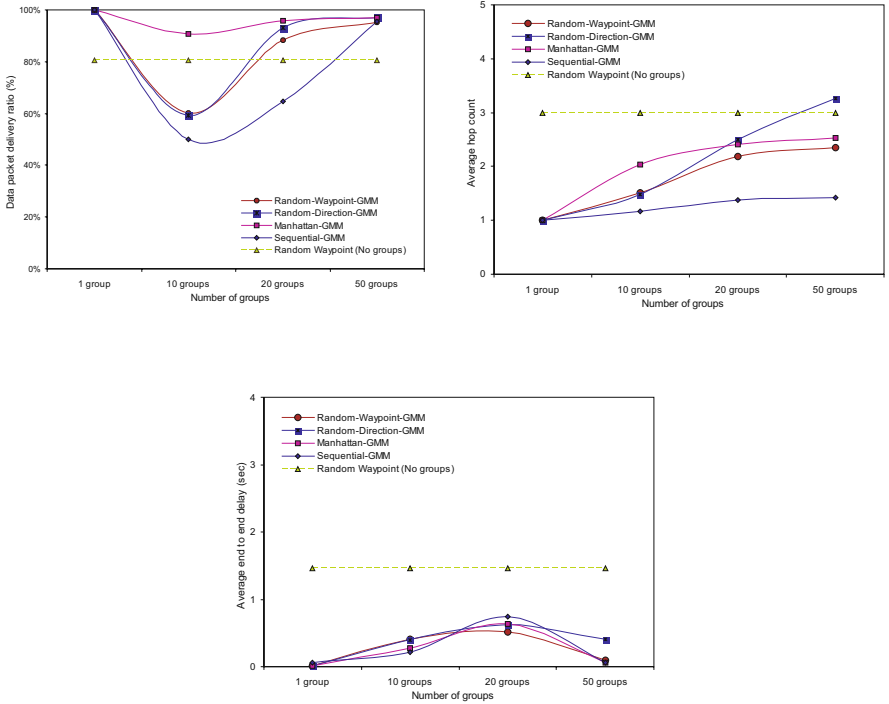


Fig. 5. Performance results for the 100 nodes MANET with five different mobility models as a function of the number of groups

4.4 Impact of the Area Size

Finally, we evaluate the impact of the simulation area size. Figure 6 shows the obtained results when varying the size of the simulation area from 500 meters×500 meters, 1000 meters×1000 meters and 2000 meters×1000 meters.

In general, as we increase the size of the simulation area all the scenarios need longer routes for routing. Longer routes also affects data packet delivery ratio and average end-to-end delay. The 500 meters \times 500 meters scenario is dense enough to make all the approaches successfully deliver around 99% of the total data packets.

As we increase the size of the simulation area, we get a quite sparse scenario specially for those scenarios using any of the group mobility models. The performance results obtained for RWG, RDG and MHG in the scenario of 2000 meters \times 1000 meters are mainly due to the effect of transient partition.

Finally, when we increase the simulation area, the SQG obtain similar performance with respect to the basic scenario. As we previously commented (see Section 4.1) the SQG suffer transient partition even when the simulation area is 1000 meters \times 1000 meters and when we increase the area its behavior remains just like in the basic scenario.

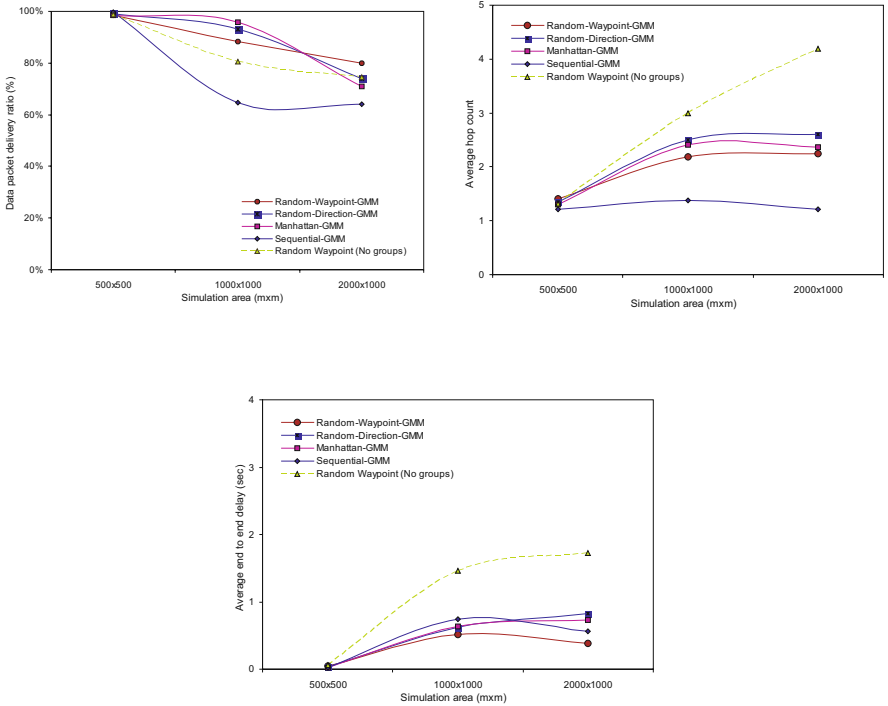


Fig. 6. Performance results for the 100 nodes MANET with five different mobility models as a function of the simulation area size

5 Conclusions

This paper presented an analysis of the behavior of a routing protocol when group mobility is involved. The objective was to prove that the chosen mobility model can deeply affect the performance results of a routing protocol. We concentrate on group mobility because there is a growing attention on the development and evaluation of MANET's approach applied to *personal area networks* (PANs), especially based on Bluetooth technology.

We proposed four different group mobility models: the *Random Waypoint Group Mobility Model* (RWG), the *Random Direction Group Mobility Model* (RDG), the *Manhattan Group Mobility Model* (MHG) and the *Sequential Group Mobility Model* (SQG). We described a group mobility patterns generator called `grcmob` whose approach is similar to that of the `setdest` module defined by CMU Monarch projects to be used with the ns-2 simulator. We compared the behavior of a classical reactive routing protocol, the *Dynamic Source Routing* (DSR) protocol and we perform a thorough evaluation of its behavior using as a reference the behavior obtained with the random waypoint model.

We observe the high variability of the results and the need to know exactly the behavior of the system and the impossibility to define a unique proposal which is general to whatever environment. We make evident that group mobility pattern highly affects the performance of a routing protocol but also that the mix of inter- and intra-group communication has a strong impact on the routing protocol performance and should therefore be taken into consideration when tuning or designing a routing protocol. Finally, the presence of groups obviously forces the network topology to be more sparse and therefore the probability of network partitions grows.

As a general rule, when intra-group data traffic ratio exceeds the inter-group data traffic the routing protocol can take advantage of group-awareness and optimize table management due to the reduced average hops count. In this context application with a lot of dependence on end-to-end delay can improve their performance due also to the high delivery ratio.

When inter-group data traffic exceeds the intra-group data traffic in general performances get worse due basically to the high sparseness of the network. Finally, the SQG is the mobility model that generally produces the worst result, especially at low speeds. In this case long duration disconnections should probably be handled at the application layer with some form of caching.

Acknowledgments. This work was partially supported by the Spanish CICYT under Grant TIC2003-00339 and by the *Junta de Comunidades de Castilla la Mancha*, Spain, under Grant PBC-03-001.

References

1. Internet Engineering Task Force, "Manet working group charter," <http://www.ietf.org/html.charters/manet-charter.html>.
2. K. Fall and K. Varadhan, "ns notes and documents.," The VINT Project. UC Berkeley, LBL, USC/ISI, and Xerox PARC, February 2000, Available at <http://www.isi.edu/nsnam/ns/ns-documentation.html>.
3. Xiang Zeng, Rajive Bagrodia, and Mario Gerla, "Glomosim: a library for parallel simulation of large-scale wireless networks," in *Proceedings of the 12th Workshop on Parallel and Distributed Simulations, Banff, Alberta, Canada*, May 1998.
4. Xiaoyan Hong, Taek Jin Kwon, Mario Gerla, Daniel Lihui Gu, and Guangyu Pei, "A mobility framework for ad hoc wireless networks," *Proceedings of the Second International Conference, MDM 2001 Hong Kong, China, LNCS Vol. 1987*, pp. 185–196, January 2001.
5. T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.
6. David B. Johnson and David A. Maltz, *Dynamic Source Routing Protocol in Ad hoc wireless Networks*, chapter 5, pp. 153–181, Kluwer Academic Publishers, 1996.
7. Promoter Members of Bluetooth SIG, *Specification of the Bluetooth System - Core. Version 1.1*, Bluetooth SIG, Inc., February 2001.
8. Mario Gerla, Rohit Kapoor, and Manthos Kazantzidis, "Ad hoc networking with bluetooth," in *Proceedings of the first ACM Workshop on Wireless Mobile Internet at MobiCom 2001, Rome, Italy*, July 2001.
9. Elizabeth M. Royer, P. Michael Melliar-Smith, and Louise E. Moser, "An analysis of the optimum node density for ad hoc mobile networks," in *Proceedings of the IEEE International Conference on Communications, Helsinki, Finland*, June 2001.
10. Christian Bettstetter, "Smooth is better than sharp: A random mobility model for simulation of wireless networks," in *Proceedings of the 4th ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM), Rome, Italy*, July 2001.
11. Ben Liang and Zygmunt Haas, "Predictive distance-based mobility management for PCS networks," *Proceedings of IEEE INFOCOMM 1999, New York, NY, USA*, March 21–25 1999.
12. Miguel Sanchez and Pietro Manzoni, "A java based simulator for ad-hoc networks," in *Proceedings of the 1999 SCS International Conference On Web-Based Modelling & Simulation, San Francisco, California, USA*, January 1999.
13. X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proceedings of ACM/IEEE MSWiM'99, Seattle, WA*, August 1999, pp. 53–60.
14. M. Bergamo, Hain R, K. Kasera, D. Li, R. Ramanathan, and M. Steenstrup, "System design specification for mobile multimedia wireless network (MMWN)," Draft, DARPA project DAAB07-95-C-D156, October 1996.
15. Karen H. Wang and Baochun Li, "Group mobility and partition prediction in wireless ad-hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC 2002), New York City, New York, USA*, April 2002.
16. Jungkeun Yoon, Mingyan Liu, and Brian Noble, "Random waypoint considered harmful," *Proceedings of IEEE INFOCOMM 2003, San Francisco, California, USA*, March 30–April 3 2003.

Activity-Based User Modeling in Service-Oriented Ad-Hoc-Networks^{*}

Tobias Breyer, Michael Klein, Philipp Obreiter, and Birgitta König-Ries

Institute for Program Structures and Data Organization
Universität Karlsruhe

D-76128 Karlsruhe, Germany

research@tbreyer.de, {kleinm,obreiter,koenig}@ipd.uni-karlsruhe.de

<http://www.ipd.uni-karlsruhe.de/DIANE/en/>

Abstract. Wireless network research still lacks methods to integratively evaluate the performance that can be expected from application layer protocols. The user behavior is predominantly affecting network performance and shows itself in two parts: its mobility and its network usage. However, it is often reduced to analytical mobility models and network traffic models separating otherwise intertwined parameters. This paper demonstrates that the use of an integrated view based on the users' real-world activity can explain network-relevant parameters both with respect to mobility and to network usage and thereby allows a more natural and realistic modeling of user behavior. The benefits are presented with the help of our graph-based mobility model and its accompanying network usage model.

1 Introduction

Wireless mobile ad hoc networks (MANETs) are a fascinating alternative to infrastructure-based cellular wireless networks. Consequently, they have been a popular research topic for the last few years. In the beginning, most of this research was focussed on making MANETs technically feasible. The major prerequisite for this was the development of appropriate routing protocols. In order to evaluate these protocols all that was needed were some more or less straightforward adaptations of existing network simulation software. Recently, however, more and more research has been dedicated to application level usability of MANETs [1,2,3]. Here, the main challenge is to enable users to access the heterogeneous resources spread across the network. To evaluate these approaches, again, a simulation is needed. However, it is not so important to find a realistic model of the underlying network. Rather, the decisive factor for the performance evaluation of application level approaches is a realistic model of the user. Unfortunately, up to now, no realistic user model exists, making it virtually impossible

^{*} The work done for this paper is partially sponsored by the German Research Community (DFG) in the context of the priority program (SPP) no. 1140.

to reliably evaluate application level approaches. Of course, users and their behaviour have been modeled in the past, however, these models have a number of drawbacks which make them unusable for the simulations needed here:

Many models assume a basically random movement of the user. While this is sufficient to simulate the performance of network level protocols, this assumption is not suitable for application level evaluation. Here, it is important to take into consideration that, usually, users do not wander around aimlessly but move in order to accomplish a certain task or a number of tasks. This implies also, that the movement of the users and their usage of the network resources are not unrelated but in the contrary highly correlated. Users will perform certain activities at certain locations exhibiting certain movement patterns. For instance, while a user may check his mail while walking from one location to another, he will most probably not edit a paper while doing so. A student may participate in an online game while relaxing in the cafeteria, but he will (hopefully) not do so while attending a class in a lecture hall. Existing models do not allow to capture this relationship, virtually all of them regard movement and network usage as two separate, independent factors.

To overcome the limitations of existing models, in this paper, we present a user-centered, integrated approach to mobility and network usage modeling which adapts to the requirements of ad hoc networks, thus facilitating a meaningful evaluation of new application protocols. Notice that our approach is applicable for other types of wireless networks, too. However, the benefits of a realistic user modeling are most obvious in the area of mobile ad hoc networks. Our activity-based approach derives an integrated view on mobility and network usage from a user's real-world activity and thereby obtains mobility patterns and service usage preferences in a natural way. Therefore, mobility is considered a secondary need that a user derives from his activities. Additionally, the needs modeled through an activity induce the service types and thereby the network usage in the mobile ad hoc network.

The paper is structured as follows: Section 2 sets the scope of the considered modeling. In Section 3 an overview of related approaches to user modeling is given, before Section 4 introduces our integrated, activity-based approach. Section 5 presents implementation issues of our user model. The paper ends with a conclusion and an outlook to future work in Section 6.

2 Scope of the Considered Modeling

In this section, we present the class of networks which is the object of our modeling: service-oriented ad hoc networks. By this means, the requirements and basic constraints of the modeling become apparent. In addition, we introduce an example from our research project DIANE: a campus scenario where students cooperate by using services with the help of mobile devices. This helps to illustrate the approach more vividly.

Service Oriented Ad hoc Networks. As introduced above, we will concentrate on modeling users in service-oriented mobile ad hoc networks. Typically, these

networks combine mobility and spontaneous membership with generic mechanisms for management and usage. For example, mobility occurs whenever people cannot accomplish their tasks at the same place during the day: Workers need to drive or walk to their workplace, to the shopping mall or to lunch, or students walk to school or move between class rooms. Mobility leads to spontaneous meetings of people where ad hoc networks allow the communication with previously known or unknown network users and the usage of the services. Other drivers may provide news about traffic, shops may provide information about their products or restaurants may inform about the meals offered. Previously known network users (co-workers or students of the same class) may provide services that are of common use.

All applications of service-oriented mobile ad hoc networks have several properties in common which lead to basic constraints and requirements of a possible modeling:

- Provision of a **diverse set of services**. This results from their mobility and the generic service paradigm in these networks. So, users with different incentives participate in the network, which leads to a high number of services.
- **Mobility and network usage are highly interwoven**. This is obvious with regard to services that are unlikely to be used in highly mobile situations, but holds true for other services as well.
- The provision and consumption of services **highly depends on the time, location and the user's need for certain services**.

Campus Scenario. To illustrate our approach, we present a university campus scenario that is characterized by a schedule-dependent mobility between locations and a wide range of possible services and network users. Here, people and institutions are able to cooperate with the help of mobile ad hoc networks. The cafeteria might be offering information about its meals to customers coming near, and co-workers or fellow students could provide useful bookmark lists. Common institutions such as libraries might be offering information about recently acquired literature. Especially on university campuses, the students' incentive to help each other by sharing knowledge with fellow students is high and thereby includes a wide range of possible ways to cooperate¹. For example, their cooperation consists of collaboratively solving open questions with the help of discussion forum services or providing useful related or summarized information about their lectures through document sharing services.

Also, mobility is an important issue on such a university campus. On university campuses, students move between lecture halls, dormitories, the cafeteria or other common institutions. Employees move between meeting halls, workplaces, the car park or the cafeteria on corporate campuses. Besides the movement between those locations, there is also motion during the stay at one of them. For example, customers in a canteen move to receive the components of their meals and finally to one of the tables.

¹ Compare with the *collective pattern* in [4,5].

In general, the set of members of a given ad hoc subnetwork constantly changes due to the movement and the different daily schedules of the people moving. However, the schedule of lectures resp. meetings leads to time periods with high mobility (lecture breaks) followed by periods with low mobility (lectures, meetings). Therefore, a highly varying network topology is followed by a rather constant one.

3 Related Work

In the literature, different user models have been proposed. Apart from user models for wireless networking, it seems promising to take a closer look at user models for travel demand modeling. This stems from the relative maturity of travel demand models. Therefore, in this section, we give a short overview of both domains of user models and conclude by pointing out the need for activity-based modeling.

3.1 User Models in Wireless Networking

We distinguish three types of user model in wireless networking. The most basic is the type of analytical mobility models that describe the movements of network users. The second type of user model adds a simple, independent network usage model and thereby represents user behavior by two separate models. The third type of user model increases the semantics of its underlying mobility model and therefore allows the integration of more sophisticated network usage models. In the following, we will analyze these types in more detail.

Analytical Mobility Models. In most user models for wireless networks, user behavior is described in terms of mobility models. Research has created a wide range of mainly analytical mobility models such as the Random Walk, the Random Waypoint [6] or the Gauss-Markov [7] model. However, their analytical properties are hindering their use for application-level performance evaluations. Mobile nodes move without actual incentive and without environmental constraints. Therefore, the oversimplified assumptions of unconstrained, erratic movement largely differ from reality.

Combined Mobility and Usage Models. The second type of user models adds simple network usage models to their mostly analytical mobility models. Such simple network usage models are typically traffic models that describe how users interact in terms of the amounts of data sent through the network. With regard to wireless telecommunication networks, this task has been achieved through the use of call models [8] by LAM ET AL. However, they introduced a network usage model that limits the set of services to phone calls and that is separated from the mobility model. Thus, the correlation of mobility and calling activities is modeled with the help of time-dependent call distributions. In service-oriented ad hoc networks, however, service usage strongly depends on location (which

services are available), network topology (which users are near) and mobility (which services are usable). Therefore, service usage is constantly changing with respect to the users' movement, distribution and service needs.

User-Centered Mobility Models. The third type of user model integrates realistic mobility models that approach mobility from a user-centered point of view. Therefore, they provide semantic information to network usage models that are built upon them. One example is the realistic mobility model presented by TAN ET AL. in [9]. In this model, mobility is the result of the users' interaction with each other. The perception of the environment and the general behavior is modeled after the principles to which animal herds or swarms adhere. This interaction principle couples the network usage model with the user mobility. However, the mobility model focuses on movements that are opposite to spontaneous and independent membership. However, the interaction principle would facilitate the coupling of the dependencies of the network usage models and the user mobility.

A more advanced user-centered mobility model is presented in [10] by SCOURIAS and KUNZ. The daily movements of users are used to evaluate the distribution of subscribers of a cellular phone network. In this regard, daily activity patterns are modeled with the help of an activity transition matrix and an activity duration matrix. Their network usage model is based on time-dependent call arrival probabilities. Thus, the increased level of semantics in the mobility model is not used. In [11], STEPANOV describes a similar approach of implementing an activity-based mobility model. The network usage model uses constant bitrate connections and therefore does not use the semantics provided by the mobility model either.

3.2 User Models in Travel Demand Modeling

Travel demand modeling is a more mature domain of user modeling than user models in wireless networks. This research topic from civil engineering uses information on population, employment, roads, public transport systems and travel behavior in order to forecast traffic on transportation systems. In the following, we discuss the two predominant types of travel demand modeling.

Trip-based modeling. Conventional traffic simulation in civil engineering uses trip-based models of travel demand [12]. The simplicity of these models implicates several drawbacks. On the one hand, trip-based models feature a step-wise procedure of generating immutable and independent trips. This prevents the implementation of complex human behavior and cooperation schemes. On the other hand, these models typically do not incorporate the time of day. Therefore, it is difficult to describe time-dependent phenomena such as congestions during rush hours.

Activity-based modeling. Activity-based approaches towards user modeling have become popular in transportation research. Thus, a number of activity-based approaches have been developed that differ in the way how activity patterns

are modeled. KITAMURA categorizes them in [13]. TIMMERMANS focuses on the scheduling of activities in the daily course of people’s life and gives an overview in [14].

Activity-based modeling typically describes daily user behavior as sequences of activities derived from a set of parameters. The respective models can be distinguished by the way they model the user’s decision of when and how an activity is carried out. One type of model describes decisions with the help of linear structure equations of a set of parameters whose dependencies are well understood. Another type imitates the cognitive processes of human decision-making by examining the activities to choose from and scheduling them according to the decision models.

These so-called activity scheduling models describe decision-making in a more natural way. One widely used type of activity scheduling models represents the human decision process as the application of a sequence of prioritized production rules. This method does not necessarily optimize the utility of an activity sequence, but allows for a dynamic and application-dependent way of determining activity patterns.

3.3 Summary

The use of analytical mobility models in wireless network research has shown similar problems as in travel demand modeling. The complexity of human behavior has been reduced to the selection of only mathematically connected movement parameters. Similarly to the trip-based models in transportation research, analytical mobility models neglect the effects of time-dependent behavior. Especially in our campus scenario, where mobility increases during lecture breaks, this is a severe drawback. The independent modeling of trips is similar to the memory-less nature of most analytical mobility models that cannot support saturation effects and sequence-dependent behavior.

In analogy to the recent developments of travel demand modeling, we conjecture the necessity of activity-based modeling in wireless networks.

4 Activity-Based Approach

Our approach to user modeling is structured in four main parts (cf. Figure 1): The **activity model** determines which activities users are performing. The **motion model** calculates the movement of the users with the help of a multigraph²-based mobility model. The **service model** realizes the network usage model. Both the motion model and the service model derive their respective parameters from the activity model. The **environment model** provides the paths that are available to the motion model and the locations where activities take place. In the following sections, we will describe these models in more detail.

² A multigraph is a graph in which every pair of nodes may be connected by several edges. In this paper, we assume that these edges are ordered.

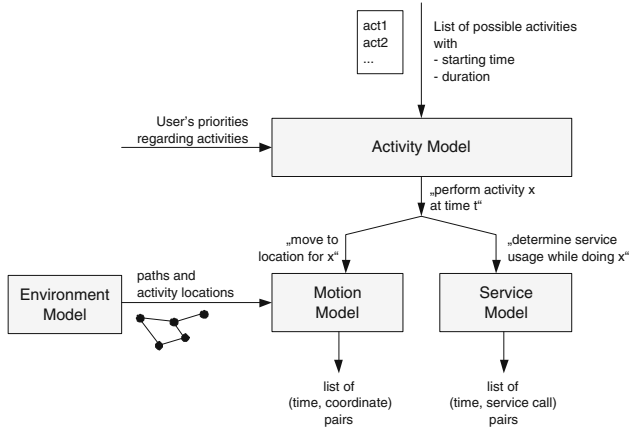


Fig. 1. The four parts of our activity-based approach. The *activity model* calculates a user's timetable of non-networking activities. This is used within the *motion* and the *service model*, which derive the necessary movements as well as useful services during this activity. The *environment model* provides the necessary information about the paths and activity location of the simulation area.

4.1 Activity Model

The general assumption of activity-based models in transportation research is that motion is not a primary need in daily life. It rather enables the execution of activities by connecting the locations of two consecutive activities. This assumption is valid in scenarios where people are not moving as an end in itself. This is typically the case in the campus scenario from Section 2.

The main task of the activity model is to transform an abstract list of possible non-networking activities into a concrete schedule of these activities. This becomes possible when activities are described with at least two characteristics: a more or less restrictive **starting time** and a typical **duration**. Additionally, each user or class of similar users weights these activities by a set of **priorities**, which helps to decide in case of conflicts. This process is described in more detail below.

The first step of creating an activity model is the identification of non-networking actions that are typical in the scenario to be considered. However, this results in an overly detailed list of actions. Therefore, actions of users are collected into classes of actions. Generally, actions from a class can be treated equally, if the desired level of detail requires no further distinction and the actions' parameters allow for a common treatment. The concept of an action class leads to the collective term of an activity. It represents the entirety of parameter sets for an action class. For example, in the university campus scenario, attending lectures and seminars, preparing for an exam, but also pursuing social activities like recreating or eating can be identified as typical actions. If a rather low level of detail is required, lectures and seminars can be generalized to an activity of

a *course*. These actions typically share common locations and are both regular, recurring events. Therefore, the subsumption of these actions is possible.

In the next step, the starting time of each activity has to be determined. Typically, we can differentiate between activities with a fixed starting time such as attending some lectures and free-floating activities like borrowing a book which can be scheduled at will except for possible constraints like opening hours.

Furthermore, the duration of an activity is important. In general, it can be fixed or variable within a certain range. Typically, free-floating activities have durations that adhere to random distributions. Therefore, the distributions and their respective parameters have to be identified.

Having identified time and duration, activities often will overlap in time. Therefore, we choose a priority scheduling technique. Hence, each user or class of users classifies the activities in pre-planned, high priority activities and spontaneous activities of lower priority. In our scenario, compulsory events such as lectures and seminars are of high priority, while free-floating activities such as the borrowing of books at the library are of lower priority. Furthermore, people are also carrying out standard activities that use up their time in between these special activities. For example, with regard to employees on a corporate campus, these non-special activities could be the steady work at their office desks. These standard activities are of lowest priority and are performed in the remaining time. As a rule, activities of higher priority take precedence over activities of lower priority when there are conflicting starting or ending times.

To summarize, activities are described by the following characteristics:

- **Starting time:** Activities either start at a fixed point in time or are freely schedulable in time.
- **Duration:** Activities last either for a fixed time or their duration is described by a random distribution.
- **Priority:** Activities are scheduled in the order of their priority when time conflicts arise. Notice that each user or each class of users defines its own set of priorities.

With all these information, the activity model is able to calculate concrete activity schedules for a user. To do that, the model tries to develop an optimal plan by placing freely schedulable activities around the ones with fixed starting time. In case of conflicts, it prefers activities with a higher priority.

4.2 Environment Model

Environment models are constraining the movements in realistic mobility models. Therefore, mobile nodes described in such mobility models can no longer move arbitrarily, but have to adhere to the constraints given by buildings, vegetation or route sections.

In our campus scenario, people are typically moving around on foot. Pedestrians' movements mostly follow common paths which exist in different widths. Therefore, our environment model describes the entirety of paths as edges in

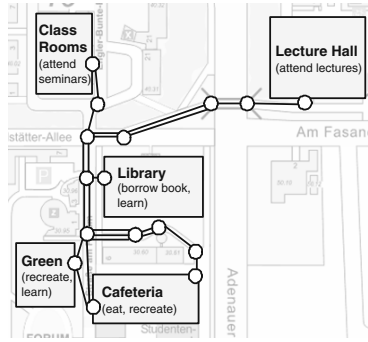


Fig. 2. Environment model of our campus scenario as multigraph. Activity locations like the library or the cafeteria (depicted by rectangles with location name and possible activities) are connected via a multigraph representing walkable paths of the real world. The width of a path is represented by the edge multiplicity.

an undirected multigraph (see Figure 2). The vertices of the graph typically represent places where several trails meet. By varying the number of vertices, the level of detail is adaptable to the needs of the application. The vertices are connected by (possibly multiple) edges that stand for the paths on the campus. The number of edges connecting a certain pair of vertices represents the width of a path. With the help of these widths, a realistic group movement where several people are moving together can be modeled. Therefore, individuals are able to move side by side on one of these parallel paths.

Additionally, the environment model describes activity locations. Besides the representation as vertices of the graph, these locations are modeled as rectangles. Each of these locations is suitable for certain activities and is characterized by its own mobility profile. Typically, analytical mobility models are suitable to describe the motion within the activity area described by the rectangles. Their use is facilitated by the rectangular shape of these areas. Generally, reaching a vertex connected to an activity location is equivalent to beginning the execution of an activity. Moreover, this involves a different type of mobility than the one connecting two consecutive activity locations.

4.3 Motion Model

The main task of the motion model is to break down the high-level movement commands (“move to a location which enables activity x ”) to fine-grained micro-movements. This can be done in three steps so that the motion model is split up in three layers: the location determination layer, the route calculation layer and the path selection layer (see Figure 3). To do its work, each layer receives certain information from the environment model. In the following, we will describe these three layers in more detail.

The highest layer is the **location determination** layer. It transforms semantic movement commands like “move to a location for activity *eating*” into

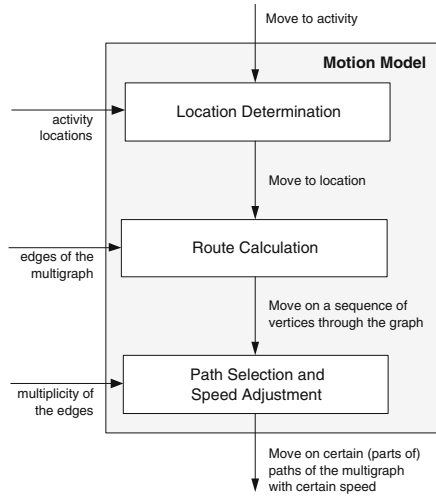


Fig. 3. The three layers of the motion model. In the *location determination layer*, a suitable location for a given activity is selected. After that, the *route calculation* determines a sequence of intermediate vertices to reach that location. Finally, the lowest layer performs a *path selection and speed adjustment* step to handle effects of crowded paths and slower moving pedestrians.

physical movement commands like “move to the cafeteria”. This becomes possible by the information about the possible activity locations, which is provided by the environment model. Generally, in this step, a random (or the nearest) possible location for that activity is chosen, but special user preferences (“For learning, I prefer going to the library”) can change this behavior.

The second layer performs the **route calculation**. It has to determine the sequence of nodes between the actual and the desired activity location. As we assume that pedestrians optimize their movements more or less, we calculate the shortest path between these two nodes. Again, the information about the graph is offered by the environment layer.

Finally, the lowest layer executes the **path selection** and the **speed adjustment**. This is necessary as pedestrians are not alone but have to deal with overcrowded paths. Therefore, users are selecting one of the parallel neighbored paths distributing themselves over the width of the available walking space. Nonetheless, this can lead to paths being overcrowded. Therefore, in our model, the speed of a user depends on the number of users walking on a path. Thus, if a lot of users have chosen to use a certain path, this will reduce their walking speed on this overcrowded path. To reduce complexity, this speed adjustment affects all current users of a certain path.

As a final result, the motion layer provides a way through the multigraph together with the according speed. According to the needed simulation granu-

larity, the bypassed coordinates can be calculated as a list of (time, coordinate) pairs. These can be used directly to determine the movement of the user.

Notice that the motion model only calculates the movements *between* activity locations. When the desired location is reached, the activity is carried out resulting in an activity-specific movement (like wandering through the shelves of a library), which is described by a specific, mostly analytical motion model.

4.4 Service Model

The service model represents the implementation of a network usage model. With the help of the semantic information available from the activity model (i.e. which activity is to be performed at what time), the service model can derive the type of service that is most useful during this activity.

We have identified four types of services that are typical in a campus scenario. These are as follows:

- **Document Services:** Static documents that are downloadable from other users in the ad hoc network. This type of service is typically provided by most users. Using this service leads to a few simple interactions.
- **Cooperative Work Services:** Services that facilitate cooperative work and induce complex interactions. Due to transactional requirements, this type of service is not available all the time or is not provided redundantly.
- **Interactive Services:** Services that induce a high number of simple interactions. As an example, chat services (instant messaging, IRC) are considered interactive services. Due to their simplicity, they are often available.
- **Dependent Services:** Services that are used only in connection with other services. For example, services that allow the printing of documents and therefore require the prior usage of a document service are considered dependent services.

We assume that service usage is dependent on a set of parameters: First, service usage depends on the **ongoing activity**. For example, the usage of document services such as additional lecture material is more useful during lectures than during lunch. Second, service usage depends on the general situation or **state of the service user**. For example, students preparing for their exams might rather use document services than interactive services. Third, service usage depends on the **level of attention** that a user can give to the usage. Therefore, we assume that faster-moving users will reduce service usage with respect to services that require a high level of attention.

We are modeling these dependencies by mapping parameter ranges to a set of service profiles. These service profiles abstract from the usage intensity of a service. The usage intensity is based on the number of interactions per unit of time and the duration of a single interaction.

4.5 Discussion

Our activity-based approach was designed with the goal to meet the new requirements of user modeling in service-oriented mobile ad hoc networks. In the following, we will examine whether these requirements have been met.

High-Level Semantics: The needs of application-level simulations in MANETs with regard to semantic information are broadly supported. Therefore, our activity-based user modeling introduces an incentive to otherwise aimless behavior. Finding simulation representations of real world scenarios is facilitated as the semantic information of a behavior's incentive is available. With respect to application-level protocols, this allows to build performance evaluations that closely fit their anticipated application scenarios and give more accurate results.

Diverse Set of Services: The spontaneous arrival, mobility and membership of users in ad hoc networks induces a potentially diverse set of available services. Therefore, our approach to user modeling handles network resources in their diversity. Thus, the activity concept allows to distinguish service usage in a natural way by an activity's main parameters, i. e. location and time. Service diversity often results from the diversity of users.

Interwoven Mobility and Network Usage: Our activity-based approach allows dependencies between mobility and network usage. Therefore, it derives both mobility and service usage from user activity. Our approach maps parameters concerning mobility, activity and personal situation to appropriate usage profiles. Hence, it is possible to examine the effects of mobility on service usage and thereby network performance.

Dependencies on Time, Location and User's Needs: Our approach captures these dependencies by correlating the user's current activity and his service usage in the service model.

5 Realization and Evaluation

Our user model has been realized as a set of meta protocols using our high-level network simulator DIANEmu [15] (see screenshot in Figure 4). With the help of a layered protocol approach, the visualization and protocol implementation have been facilitated. In addition to the protocols implemented for the activity, motion and service models, the implementation of our approach features a user model. While the former models are implemented as protocols that are interacting locally, the user model allows the cooperation of nodes in the network. Therefore, conjointly planned activities are possible.

In spite of the integration into our simulator tool, the mobility and service usage data is not bound to it. We are separating this data from its application to network protocol performance evaluation. This allows for reproducible results when used in other simulation environments.

Realistic user models need input data to imitate actual users in the scenario evaluated by simulation. Therefore, the environment model is based on the university campus in Karlsruhe. For the activity data, we use actual lecture

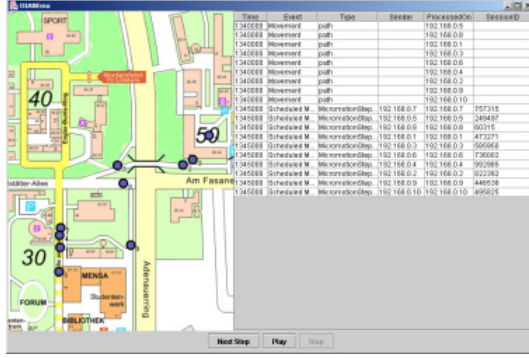


Fig. 4. Simulator DIANEmu View. Left: Visualization of users moving on campus. Right: Message view of the underlying meta protocols.

schedules, opening hours of campus institutions and survey data. Therefore, we conducted a survey [16] about students' activities and service usage on the campus. The data used in our model is based on the answers of about 80 first-year students in computer science. The survey provides us with data about the students' activity and service priorities as well as their activities' frequency and duration. Additionally, it gives us data about the dependencies of the service usage and its parameters such as mobility or personal situation.

For evaluation, we are comparing the mobility from our activity model with the motion of students as stated in our survey. Therefore, our survey featured questions about the students' daily course of activities. This included information about the students' location at several points of time during the day.

The survey data is currently being used to evaluate the level of reality of our model. Therefore, we compute the deviation of the users' positions derived from our model with the reference positions given by the survey data. A user's activity motion is then matched to that survey participant's data giving the smallest error sum of squares. For comparison, this matching is also done for a graph-based Random Waypoint implementation using appropriate wait times.

6 Conclusion and Future Work

In this paper, we have motivated the need for realistic, semantically rich user models for service-oriented ad hoc networks. Then, we have presented an activity-based approach from transportation research that derives user mobility from the basic human need to carry out activities. With the help of this semantic model, we have presented our integrative approach of a graph-based mobility model and a semantic network usage model.

Currently, we are refining the modeling of free-floating activities and sequence-dependent services. In the future, we will evaluate our application level protocols with the help of the mobility and network usage data derived from the activity model.

References

1. Klein, M., König-Ries, B.: Multi-layer clusters in ad-hoc networks – an approach to service discovery. In: International Workshop on Peer-to-Peer Computing, in the course of Networking 2002 Conference. (2002)
2. Klein, M., König-Ries, B., Obreiter, P.: Service rings – a semantic overlay for service discovery in ad hoc networks. In: 14th International Conference on Database and Expert Systems Applications DEXA'2003, Prague, Czech Republic (2003)
3. Klein, M., König-Ries, B., Obreiter, P.: Lanes - a lightweight overlay for service discovery in mobile ad hoc networks. In: 3rd Workshop on Applications and Services in Wireless Networks (ASWN2003), Bern, Switzerland (2003)
4. Obreiter, P., Nimis, J.: A taxonomy of incentive patterns – the design space of incentives for cooperation. In: Proceedings of the Second International Workshop on Agents and Peer-to-Peer Computing (AP2PC'03), Melbourne, Australia (2003)
5. Obreiter, P., König-Ries, B., Klein, M.: Stimulating cooperative behavior of autonomous devices – an analysis of requirements and existing approaches. Technical Report 2003-1, Universität Karlsruhe, Faculty of Informatics (2003)
6. Johnson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless networks. In Imielinski, Korth, eds.: Mobile Computing. Volume 353. Kluwer Academic Publishers (1996) 153–181
7. Liang, B., Haas, Z.J.: Predictive distance-based mobility management for PCS networks. In: INFOCOM (3). (1999) 1377–1384
8. Lam, D., Jannink, J., Cox, D.C., Widom, J.: Modeling location management for personal communication services. In: Proceedings of 1996 IEEE International Conference on Universal Personal Communications (ICUPC96). (1996) 596–601
9. Tan, D.S., Zhou, S., Ho, J.M., Mehta, J.S., Tanabe, H.: Design and evaluation of an individually simulated mobility model in wireless ad hoc networks. In: Communication Networks and Distributed Systems Modeling and Simulation Conference 2002, San Antonio, TX (2002)
10. Scourias, J., Kunz, T.: An activity-based mobility model and location management simulation framework. In: Proceedings of the Second ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM'99), Seattle, USA (1999) 61–68
11. Stepanov, I.: Integrating realistic mobility models in mobile ad hoc network simulation. Master's thesis, Universität Stuttgart, Fakultät Informatik (2002) Thesis Nr. 1989.
12. Stopher, P., Lisco, T.: Modeling travel demand: A disaggregate behavioral approach, issues and applications. In: Transportation Research Forum Proceedings. (1970) 195–214
13. Kitamura, R.: Applications of models of activity behavior for activity based demand forecasting. In: Activity-Based Travel Forecasting Conference Proceedings. (1997)
14. Timmermans, H.: Models of activity scheduling behaviour. In Beckmann, K.J., ed.: Tagungsband zum 2. Aachener Kolloquium 'Mobilität und Stadt' (AMUS 2001), Institut für Stadtbauwesen und Stadtverkehr, Rheinisch-Westfälische Technische Hochschule Aachen (2001) 33–47
15. Klein, M.: Dianemu – a java based generic simulation environment for distributed protocols. Technical Report TR 2003-7, ISSN 1432-7864, Institute for Program Structures and Data Organization, Universität Karlsruhe (2003)
16. Breyer, T.: Online survey on students' activities and service usage on campus (2003) <http://www.tbreyer.de/umfrage.html>, in German.

Media Access Control Schemes for Mobile Ad Hoc Networks

Chun-Hung Lin and Chien-Yuan Liu

Department of Computer Science and Engineering
National Sun Yet-Sen University, Kaohsiung 804, TAIWAN
lin@nssysu.edu.tw, cyliu@cse.nssysu.edu.tw

Abstract. Multi-rate capabilities are supported by the physical layers of most 802.11 devices. To enhance the network throughput of MANETs, transfer rate adaptation at MAC layer should employ the multi-rate capability at physical and the information of previous transmissions provided by MAC and physical layers. In this paper, we propose a transfer rate adaptation scheme plus back-to-back frame transmissions, and fragmentation at MAC layer, named TRAF. TRAF adopts a bi-direction-based approach with an extended option to select an appropriate rate for frame transmission under fast changing channel conditions. Consecutive back-to-back frame transmissions to fully utilize good channel quality during a coherent time interval and fragmentation algorithm to maintain high throughput under worse channel conditions are recommended in TRAF. Extensive simulation is experimented to evaluate the performance of TRAF. Regarding simulation results, frame delivery ratio, and network throughput of TRAF are significantly improved by comparing to that of other protocols.

1 Introduction

Wireless local area networks (WLANs) are becoming increasingly popular [10]. New requirements of high speed transmissions for broadband wireless multimedia applications are emerging. Scientists and engineers are developing and designing efficient modulation techniques and media access control schemes to accomplish the requirements. Nowadays, the IEEE 802.11 [1, 2, 3] MAC protocols can provide physical-layers with multi-rate capabilities. In original 802.11, a frame is sent at a single base rate. With the multi-rate capability, a number of rates can be chosen to transmit frames according to channel conditions.

In mobile wireless networks, path loss, fading, and interference cause variations in the received signal-to-noise ratio (SNR) and the bit error rate (BER). Because of the lower the SNR, the more difficult it is for the modulation to decode the received signal. Since high rate transfer typically applies denser modulation encodings, a trade off generally emerges between data rate and BER. When SNR is sufficiently high to switch to a higher speed, such that BER of the modulation still be preserved under an acceptable level for correctly encoding the received data, the higher speed modulation can be selected to transmit frames at a higher rate.

Some control protocols for rate adaptation are proposed. ARF [14] is a sender-based approach. RBAR [11] recommends that a sending rate is selected by a receiver instead of by a sender. OAR [18] indicates that the channel quality is stable in a channel coherent interval. A sender can exploit the interval to send more frames. In this paper, we propose TRAF protocol. TRAF provides an extended option, a two-way rate adaptation scheme and per-frame-based back-to-back transmissions, and frame fragmentation algorithm. To evaluate the performance of TRAF, a simulator is developed to measure frame delivery ratio and network throughput of TRAF. Simulation results are compared to that of other rate adaptation protocols. The results show that TRAF outperforms other approaches.

The remainder of this paper is organized as follows. In Section 2, we describe some background material concerning 802.11. Related researches, e.g. ARF, RBAR, and OAR are briefly presented in Section 3. In section 4, we firstly point out our observations about the issues in previously proposed approaches. Then, we propose our solutions to the issues. Performance simulation and discussion are presented in Section 5. Finally, Section 6 concludes this paper.

2 IEEE 802.11 Overview

A brief introduction of the IEEE 802.11b based on frequency hopping spread spectrum (FHSS) physical layer is described in this section. The description includes DCF, NAV update mechanism, multi-rate capability, and fragmentation scheme. Since the paper focuses on MANET (also known as independent basic service set, IBSS), thus only fields concerning IBSS are presented. The concept is possible to apply to other high speed 802.11 MAC.

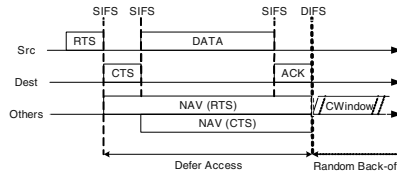


Fig. 1. DCF frame sequence

Distributed Coordination Function (DCF). In 802.11 [1, 6, 7, 9, 15], DCF is a basic medium access protocol. All traffic uses immediate positive ACK frame. The virtual carrier sense mechanism is achieved by distributing reservation information. The exchange of RTS/CTS frames prior to the actual data frame is to distribute this medium reservation information. The RTS/CTS frames contain a duration field for reserving the medium. All nodes within the reception range of either the source or the destination shall learn of the medium reservation. For example the transmissions shown in Figure 1, when Src has a frame to send, it calculates the time needed to transmit CTS, data and ACK frames (assume SIFS time is included in each frame) at current data rate, which forms the reservation duration in RTS frame. Then, Src sends the RTS frame to Dest at base rate. Others in the reception range of Src postpone their

transmission attempts for the duration declared in the RTS frame. If Dest can receive the RTS frame, it immediately replies a CTS frame with the duration field copied from RTS. Others in the reception range of Dest defer their access for the duration declared in the CTS frame. DATA and ACK frames also carry a duration field for a period to the end of ACK frame, respectively, to information hidden/exposed nodes [19].

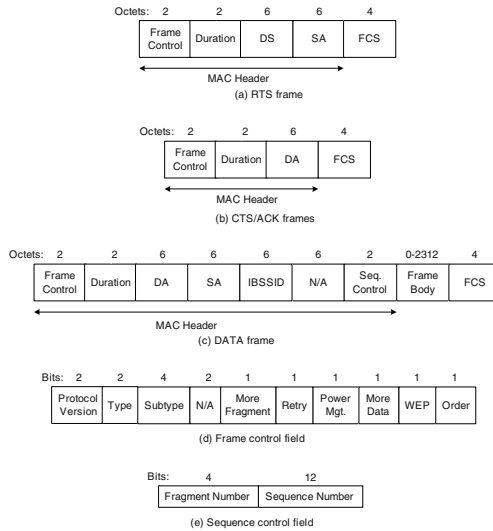


Fig. 2. Frame formats

MAC Frame Formats. There are 3 types of frame, including management frame, control frame, and data frame. In each type of frame, there are several subtypes. For example, RTS/CTS frames are subtypes of control type. The frame formats of RTC, CTS, DATA, and ACK frames are depicted in Figure 2, where SA and DA denote Src and Dest MAC addresses, respectively. More fragment bit is used for frame fragmentation and presents that there is a fragment next to current fragment transmission. Retry bit is set if the frame is a retransmitted one. Power management bit indicates a sender is working in power saving mode. More data bit represents that there is an impending frame to transmit. WEP bit means wire equivalent privacy. Frame ordering is controlled by Order bit. Fragment number and sequence number are 2 sub-fields in sequence control filed. Readers can refer to 802.11 [1] for detailed application of these fields.

Network Allocation Vector (NAV). A virtual carrier sense mechanism is referred to as the NAV. NAV maintains a predication of future traffic on the medium based on duration information that is announced in RTS/CTS frames. NAV can be thought of as a counter, which counts down to zero at a uniform speed. When the counter is zero, the virtual carrier sense indication is that the medium is idle. Nodes receiving a valid frame shall update their NAV with the information received in the duration field, but only when the new NAV value is greater than the current NAV value and the frame is not addressed to the receiving node.

Fragmentation. MAC layer may fragment and reassemble a frame. The length of a fragment shall have same octets except the last. When a frame is transmitted with fragmentation, MAC layer shall set more fragment bit, reset fragment number for the first fragment. Then, MAC layer keeps the values of more fragment and sequence number unchanged, and increases fragment number by one for each following fragment. MAC shall reset more fragment bit for the last fragment to inform the recipient to reassemble all the received fragments. Retransmission is allowed for the fragmentation and de-fragmentation mechanisms.

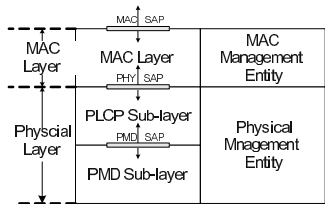


Fig. 3. Layer model

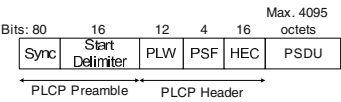


Fig. 4. PLCP frame format

Multi-rate Capability. Nowadays, physical layers in most of wireless interfaces have multiple data transfer rate capabilities that allow MAC management entity to perform dynamic rate switching with the objective of improving the performance. All control frames and frames for broadcast or multicast shall be transmitted at base rate, which is one of rates in the IBSS basic rate set and is determined on the network started, so they will be understood by all nodes. Data and/or management frames with unicast address shall be transmitted on any supported data rate selected by rate switching mechanism.

A physical layer is divided into physical layer control protocol (PLCP) and physical medium dependent (PMD) sub-layers. Figure 3 shows the 802.11 layer model. A MAC frame, also named MAC protocol data unit (MPDU), with rate information is passed a parameter of physical service access point (PHY_SAP) to the PLCP sub-layer. In PLCP sub-layer, a PLCP preamble is added in the front of PLCP header followed by a payload, which is the MPDU passed from the MAC layer. The rate parameter in PHY_SAP call is formatted into the payload rate information in the PLCP header. The format of PLCP frame is shown in Figure 4. The PLCP preamble and PLCP header shall be transmitted at the base rate chosen from the IBSS basic rate set. The PLW specifies the number of octets contained in the PSDU. The PLCY payload will be transmitted at the rate specified in the PLCP signaling field (PSF) as shown in Figure 4. Therefore, all receivers can synchronize to the PLCP preamble and the recipients can receive the PLCP frame at the same rate used by the sender.

3 Related Work

In wireless communications, the signal at the recipient is a superposition of different reflections of the same signal, noises of background and interferences from nearby

channels. The received signal power strength is heavily dependent on the spatial locations of sender and recipient. Any motion of sender or recipient causes the signal strength to vary with time. SNR is used by the physical layer to capture the channel [17]. The larger SNR, the better the chance of any frame is received with lower BER. In addition, some modulation schemes are more efficient than others at transmitting information with a given SNR and bandwidth [8]. Currently, almost all physical interfaces for WALN provide multiple transfer rates that support dynamic rate switching for improving the throughput of WLAN. There are two steps to rate selection: Channel quality estimation and rate adaptation. Channel quality estimation involves measuring the time-varying metrics (e.g. SNR, BER) of wireless channel which would be used as a short-term or long-term predictor. Rate adaptation schemes employ the predictor to select an appropriate rate. A threshold-based scheme in [5, 13] is a common technique for rate determination.

Some rate adaptation protocols [4, 11, 12, 14, 16, 18] have been developed for MANET. In [14], the authors propose the auto rate fallback (ARF) protocol for 802.11 used in Lucent's WaveLAN II devices. In ARF, the sender selects the best rate based on information from previous data frame transmissions. It incrementally increases or decreases the rate after a number of consecutive successes or losses, respectively. In this way, ARF doesn't need to modify 802.11 MAC frame formats. But the previous information can't represent current channel condition at present.

In [11], the authors present the receiver-based auto-rate (RBAR) protocol. The key idea of RBAR is to allow a receiver to estimate channel quality and to select an appropriate rate during RTS/CTS frame exchange for the next data frame. Since the rate selection is done by a receiver during latest RTS/CTS exchange, the channel quality estimation is nearer to actual transmission time of data frame than the sender-based approaches, like ARF. However, the rate chosen by the receiver must be carried back to the sender by CTS frame. Modification to 802.11 MAC frame formats to carry the rate related information is then unavoidable. RBAR redefines both MAC layer and physical layer frame formats. At MAC layer, RBAR redefines the 16-bits duration fields of RTS/CTS frames into two sub-fields: 4-bits data rate and 12-bits data frame length. The purpose of the modification is a tentative reservation to inform all overhearing nodes to calculate and to set their NAV by the rate and the length information. A new reservation sub-header (RSH) is also redefined into the data frame header as a final confirmation to previous tentative reservation. At physical layer (PLCP), the signal field of PLCP header is redefined to be 4-bits RSH rate and 4-bits data rate. RBAR applies different rates to send RSH and data frame.

Sadeghi et al. [18] propose another receiver-based approach, the opportunistic auto rate (OAR) MAC protocol. The key observation is that channel coherent times are typically at least multiple frame transmission times. Consequently, when a mobile sender is granted channel access while encountering a high quality channel, OAR grants the sender a channel access time that allows multiple frame transmissions in proportion to the ratio of the achievable data rate over the base rate. Consequently, OAR transmits more frames under high quality channel than under low quality channel. OAR basically appears similar features and drawbacks of RBAR, such as timely receiver-based rate selection and frame format redefinition. In addition, OAR provides new back-to-back frame transmissions by using fragmentation technique to fully utilize the high quality channel for sending more frames. During back-to-back transmissions, a sender sets the more fragments flag in frame control field of MAC header for each intermediate fragment (a frame is treated as a fragment) until the last

fragment is transmitted. The sender must also set the fragment number in sequence control field of MAC header to 0. This prevents the receiver from treating the data fragment as a part of an actual fragmented frame. Since OAR redefines application meaning of fragmentation, it causes certain side-effect. We explain this in detail in the observation 6 of next section.

4 The Proposed Protocol

4.1 Observations

With 802.11 [1, 2, 3], most of wireless interfaces nowadays support multiple data transfer rate capabilities. As described in section 3, lots of rate switching protocols had been proposed to exploit the multiple rate capability to improve the performance of MANET. The enhancement shown in [11] was significant in comparison with that of merely using a single fixed rate.

***Observation 1:** Adequate rate adaptation would result in positive effect on the throughput of a MANET with the interfaces of multi-rate transfer capability.*

With simpler ARF, the sender heuristically selects a data transfer rate. ARF works well when the channel condition is relative smooth. Frames are easily got loss as channel variation is drastic, and the throughput of ARF clearly degrades as moving speed increased. [11] shows that ARF can not rapidly react to the fast channel variation. In contrast to ARF, RBAR decides a transfer rate by a recipient. A recipient gets the receiving condition from its radio [1]. Thus, the recipient is the most suitable one to choose a transfer rate for the following frame transmissions. The results of performance evaluation in [11] showed that BRAR outperforms ARF in considering channel variation and various moving speed.

***Observation 2:** A recipient can provide more accurate information of channel variation than a sender.*

OAR [18] adopted the concept of the receiver-based approach to quickly react a data transfer rate to a channel variation. In addition, they leverage the back-to-back fragmentation scheme to further enhance the throughput of MANET. The enhancement is mainly attributed to diminish back-off times and contention chances between several consecutive data fragment transmissions in the residual time saved with a faster transfer rate relative to the original slower one. The throughput advance of OAR against RBAR can be clearly seen in [18].

***Observation 3:** Using the residual time saved from faster transfer rate to send more frames with back-to-back frame transmissions can further enhance the throughput of MANET.*

OAR transmits all consecutive fragments with a fixed transfer rate decided by the recipient. However, to adapt the receiver-based transfer rate by per fragment basis can closely react to the latest channel condition. Thus, better channel adaptation quality than that of OAR can be obtained. This enhancement can be seen from the result of simulations in section 5.

Observation 4: *The rate adaptation to channel variation by per fragment basis is timely than a single fixed rate for all consecutive fragments.*

In RBAR, the MAC header and the PLCP header are redefined to carry the data transfer rate and to declare the data frame length. The recipient basically follows the rate specified by the sender for most of cases. However, in the case of channel condition changes violently, the receiver must refer the receiving signal level indicated by the physical layer. With the receiver-based rating concept, although the throughput is really improved by RBAR protocol, there are some drawbacks occurred. Firstly, the modifications at the duration field of the MAC header in the RTS/CTS frames. The overhearing nodes which obey only the standard definition would not interpret the rate and the length of a RBAR frame in a right way. Furthermore, the redefinition at the signal field of the PLCP header would results that the receiver can not differentiate the PLCP header of RBAR from that of standard 802.11. Therefore, RBAR can not be understood by the nodes implemented according to 802.11. OAR adopts similar redefinitions as RBAR. Thus, the drawbacks also accompany OAR.

Observation 5: *To modify the definition of MAC header and PLCP header should be very careful to insure the compatibility to 802.11 and to guaranty the interoperability to huge existing installation bases.*

OAR utilizes fragmentation to implement back-to-back fragment transmissions. Although, this is a good idea, to consider the conformation to 802.11, the back-to-back fragment transmissions may result in some side effects. Regarding 802.11, a MPDS at the sender side is possible to be fragmented in order to be sent via a wireless channel, at the receiving side all the fragments will be reassembled according to their fragment number with the same frame sequence number. In the case of two or more separated MPDSs are sent as consecutive fragments with same frame sequence, then they would be reassembled into a single MPDS at the receiver. This may cause frame disorganization, e.g. two frames belonged to two flows respectively could be combined into one frame for one flow. Another case is that the consecutive fragments (frames) have different frame sequence numbers, of course at the receiver end they can be kept for reassembling in different buffer lists. However, there is only one last fragment with more fragment bit set to zero to inform the completion for reassembling. Other frames in fragment form still wait to be reassembled until time-out occurred.

Observation 6: *The concept of back-to-back frame transmissions is nice. Yet to implement the idea with fragmentation may mislead into incompatibility and cause severe resource wastage.*

4.2 Our Methods

To get rid of the problems originated from the related research proposals as described in the observation section, we try to propose a new wireless medium access control protocol to provide not only a dynamic multi-rate adaptation capability, but also with an active fragmentation to preserve a high data transfer rate under a short-term BER rising over a intermittently interfered wireless channel. In general, all the recommendations obtained from previous observations would be infused into the new

design. However, the purpose of the utilization of fragmentation is entirely different between our method and OAR. We will explain it in detail at the back-to-back transmissions subsection later.

Option for Bi-directional Rate Adaptation. To provide a better dynamic multi-rate data transfer over an unreliable wireless channel, the receiver-based rate-decision strategy is adopted in our design. Both of RBAR and OAR approach the strategy by redefining the duration of MAC header to carry the data transfer rate. The drawback is that the newer protocols would be incompatible to huge existing 802.11 adapters. Inspecting 802.11 as Figure 2, there is no reserved field in RTS/CTS frames to carry rate information.

Instead of redefinition, we append an option field to the standard RTS/CTS/Data/ACK (abbreviated as RTS-R/CTS-R/Data-R/ACK-R) frames as depicted in Figure 5 to piggyback extension information. The option field is consisted of code and data fields. Code specifies the option type. Data is 2-byte field and its meaning is dependent on the code field. For a RTS-R, the code is 1 and the data consists of a 1-byte transfer rate, which will be used for the next transmission, and a 1-byte receiving rate at which its transceiver affords to receive.

AT the beginning of a communication, a sender formats these two rates into an option field plus a checksum, and appends the option to the tail of RTS frame. Since the RTS is sending at base rate, the overhearing nodes can understand the frame and would pick out the duration to update its NAV, and they simply ignore the tail option field. A receiver with the option capability verifies the correctness of RTS-R option by comparing the checksum at the tail and interprets the tail option back into the correct rates. A best-fit acceptable rate from an operational rate set is chosen by the recipient with referring to the rate requested by the sender and the signal level indicated from its physical layer. The best-fit acceptable rate can be chosen by a threshold-based technique as studied in [5, 13]. Let θ_i , $i=1, 2, \dots, N$, represents a SNR threshold at which the corresponding BER is equal to an adequate working level, e.g. 10^{-5} . Also let α_i represents a data transfer rate approximately matching to the threshold. Thus, a transfer rate α_i can be chosen by the following algorithms, $i=0, 1, \dots, N$, where α_0 is the minimum speed at which a transceiver can support and α_N is the maximum speed.

If (SNR < θ_1) then choose α_0
If (θ_i < SNR < θ_{i+1}) then choose α_i
If (θ_N < SNR) then choose α_N

The recipient then replies a CTS-R frame with code being 1 at base rate back to the sender. The nodes covered by the receiver will update their NAV with the duration field of CTS-R frame as it received a normal CTS frame. As our designation, the sender can get the confirmed rate from option field and will use it for the next transmission. With RTS-R/CTS-R exchange, all the nodes of conforming either to 802.11 or the option-extended one can interoperate to each others.

An alternate suggestion to add the option extension is to allocate reserved subtypes in the frame control field of MAC header of a control-type frame as shown in Figure 2(d), for defining new RTS-R/CTS-R/ACK-R control frames. In this way, the option

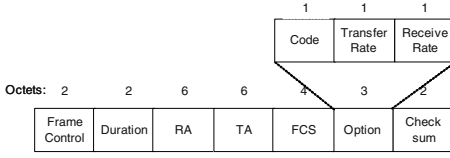


Fig. 5 (a). RTS-R format

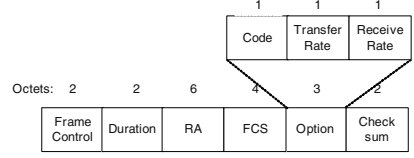


Fig. 5 (b). CTS-R and ACK-R format

filed can be put between the MAC address fields and the FCS field, and checksum field can be omitted. The sender and the recipient must support the new defined control frames to correctly interpret the option. As for the overhearing nodes, they treat the new control frames as normal control frames and extract the duration field from the MAC header to update their NAV as usual.

In summary, the observations 1, 2, and 5 in the former section are deliberately considered and designed into our proposal. Therefore, the compatibility and interoperability can be maintained in addition to support the bi-directional multi-rate frame transfer over an unreliable wireless channel by per frame basis.

Back-to-Back Transmission. The benefit of back-to-back transmissions was already explained in the observation 3 in the former section. However, the observation 6 also presented a caution of incompatibility if the back-to-back transmissions were implemented with fragmentation manner as the approach proposed in OAR. Thus, we design back-to-back transmissions with per frame basis. Consider that the time period to transmit a frame with base rate, selected from one of the basic rate set [1], is t_{FB} . Assume that the multi-rate capabilities are supported by both of a sender and a recipient. As the example with back-to-back frame transmissions shown in Figure 6, the data frame 1 which original would be sent by base rate (here we assume the base rate is 1 Mbps), now will be sent by 5 Mbps. If the sender still has other frames waiting to be sent, and the residual time, denoted as t_{FBR} , is at least long enough to transmit more than one frame (plus the length of an ACK time). The sender can transmit the next frame after SIFS when ACK-R is received. With this similar way, all the pending frames can be sent until t_{FBR} is too short to transmit the next one frame. On the other hand, suppose that there is no more frame pending at the sender and t_{FBR} is longer than a threshold ($2SIFS + RTS \text{ time} + CTS \text{ time} + C$, C is a constant), CF_End control frame is sent to inform all overhearing nodes to reset their NAV for better channel utilization. If t_{FBR} is less than the threshold, the sender will do nothing and just let the residual time pass by.

The 3-tuple specified in each frame in Figure 6 denotes current transfer rate, expected transfer rate, and acceptable receiving rate, respectively. The first rate can be specified at the PLCP signal field as shown in Figure 4. The second and the third rates are indicated on the data part of RTS-R/CTS-R/ACK-R option filed as shown in Figure 5. For instance, SRC sends RTS-R with (1, 11, 1) 3-tuple, the first number means that RTS-R is to be transferred at base rate, the second rate describes that the next frame is expected to be transferred at 11 Mbps, and the last rate represents an acceptable receiving rate at 1 Mbps which SRC can afford to receive. Next, DEST replies CTS-R with (1, 11, 5), which means CTS-R is to be transferred at 1 Mbps as

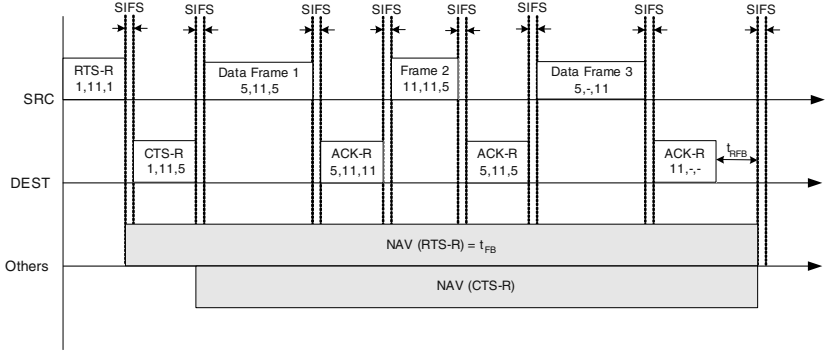


Fig. 6. Back-to-back transmission

requested by the third rate of RTS-R, and it also expects to transfer the next frame at 11 Mbps and it may receive a frame at a speed of 5 Mbps. After that, SRC transmits its first Data-R frame at 5 Mbps as the rate specified on the third rate of 3-tuple in CTS-R, expects its next transfer rate to be 11 Mbps, and expresses its affordable receiving rate to be 5 Mbps. Next, DEST knows to send its ACK-R at 5 Mbps, requests to send its next frame at 11 Mbps, and describes its affordable receiving rate at 11 Mbps. Then, SRC can transmit its the second Data-R frame at a higher rate of 11 Mbps, informs its next transmission speed to be 11 Mbps, and tells its affordable receiving rate at 11 Mbps. With the similar manner, consecutive data frames and control frames can be transmitted with bidirectional rate adaptation which best fit to the signal level on the bidirectional link.

Fragmentation Algorithm. In mobile wireless networks, interference, fading, and path loss result in the variation of SNR at recipients. Such variation further causes a variation in the BER. Higher data rate α_i usually causes higher BER for a given SNR. Note that, for a given SNR, a decrease in the data rate results in a decrease in BER as shown in Figure 7. Therefore, there is a tradeoff between data rate and BER. In data rate selection, as chosen by RBAR and OAR, the data rate is decreased while SNR decreased beneath a threshold θ , while the corresponding BER is higher than an adequate value for demodulation, e.g. 10^{-5} . Denotes δ as the short-term variation in SNR, and the increased BER corresponding to $\theta - \delta$ is too high for a normal frame transmission, e.g. BER is equal to 10^{-4} , but it is still low enough for a shorter frame to be transferred. For instance in 802.11b FHSS [1], the maximal length of a MPDU as a payload of PLCP frame transmission is specified as 4095 octets, which equals 32888 bits (128 bits PCLP header plus 8×4095 bits payload), and it is in a magnitude of 10^5 bits. Undoubtedly, a frame with a length of 10^5 would suffer a high broken probability when it is transferred over a wireless channel with 10^{-4} BER. Instead of simply choosing a lower data rate, a frame fragmentation scheme is applied to divide a normal frame become several short fragments. The length of the fragments is shortened to a magnitude of less than 10^4 , if the MPDU is partitioned into 10 or more fragments. Ideally, these fragments would have higher probability to pass through the wireless channel affected by the impact of δ variation in SNR. Thus, there is no need to switch down transfer rate and the throughput will be kept at the same level of the

original SNR as without the impact. Therefore, the data rate selection algorithms are refined by considering the δ variation and fragmentation.

If $(SNR < \theta_{1-\delta})$ then choose α_0
If $(\theta_i - \delta < SNR < \theta)$ then choose α_i plus fragmentation
If $(\theta_i < SNR < \theta_{i+1} - \delta)$ then choose α_i
If $(\theta_N < SNR)$ then choose α_N

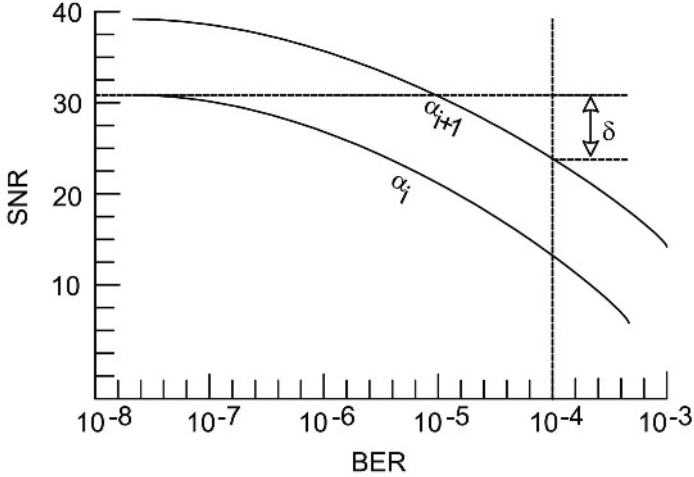


Fig. 7. SNR vs BER

Assume a bit error occurred in a normal frame transmission. The broken frame would be discarded by a recipient. The sender will try to retransmit it after a time-out timer and a DIFS period plus random back-off slots. It is still possible for a fragment being discarded by a recipient due to a bit error happened during a fragment transmission. In this case, the dropped fragment can be retransmitted with similar way. Even so, the cost of a transmitting failure for a long frame is still much higher than that of a short fragment. Let $F(i)$ and $f(i,j)$ denote frame i with length L and fragment j with length l in frame i , respectively. Also let $P(i)$ and $p(i,j)$ denote the transmitting failure probabilities of $F(i)$ and $f(i,j)$, respectively. Thus, $P(i) = 1 - (1 - BER)^L$ and $p(i,j) = 1 - (1 - BER)^l$, where $P(i) > p(i,j)$ and $L > l$. The failure cost to transmit $F(i)$ with a frame way or with fragments way would be $P(i) \times L$ and $\sum p(i,j) \times l$, respectively. With fragmentation, the cost of $P(i) \times L - \sum p(i,j) \times l$ would be saved for a frame transmission. For instance, given $L = 10^5$ bits, $l = 10^4$ bits, and $BER = 10^{-4}$, thus $P = 0.6321$, $p = 0.0952$. Therefore, with fragmentation, successful probability of a fragment is much higher than that of a frame. When one failure occurs, the saved cost is 5369 bits per frame transmission. This is also equal to throughput enhancement. Although, overhead from fragmentation is not considered here, it is relative small to compare to data fragment itself. In this example, total overhead is $9 \times (ACK \text{ time} + SIFS)$.

5 Simulation Experiments

5.1 Simulation Model

To simulate all protocols, we develop a simulator with sophisticated functions which implement detailed control schemes of each rate-adaptation protocol, including fix rate at 1, 2, 5.5, and 11 Mbps, ARF, RBAR, OAR, and TRAF. We are currently interested in the performance of proposed protocol at MAC layer viewpoint. Thus, we assume all nodes are in the transmission range of each others. In the simulator, we assume all communicating pairs are modeled as flows with constant bit rate (CBR) and send the frames with fixed frame length. Each flow is interfered with fast changing channel noise. Thus, its receiving strength is modeled as dynamically changing SNR. The SNR is randomly generated during the simulation period with a mean value of 17 dB and a deviation value of 12 dB. It is generated at a timing of exponential distribution with a mean interval of 10 ms. Moreover, mobility affects both line-of-sight interference and channel coherence time. It is modeled as a drifting effect to the SNR of each flow. When the end nodes of a flow move closer, its SNR would be increased a certain amount. On the contrary, the SNR would be decreased. We model 4 relationships corresponding to 4 data rates into the simulation model. Acceptable SNR ranges of 1, 2, 5.5, 11 Mbps transfer rates are given by [5...9], [11...15], [17...21], [23...27], respectively. BER range corresponding to SNR range at each transfer rate is given by $[10^{-4}...10^{-8}]$. For instance, at a certain time, if a flow transmits a frame with SNR equal to 19, according to the given SNR range and the given BER range, the data rate is 5.5 Mbps and the BER is 10^{-6} . As for the hidden terminal problem [19], it can be deemed as a noise impact with an abruptly changing SNR, it is implicitly included in the dynamically generated SNR values. When we simulate TRAF with fragmentation, δ variation in the fragmentation algorithm as described in Section 4 is set to 1 dB to observe throughput improvement.

5.2 Frame Delivery Ratio

The frame delivery ratio is denoted as the aggregate number of successfully sent frames divided by the aggregate number of generated frames. In this simulation, we set the CBR of each flow to 384 Kbps, the frame length is fixed to 2048 bits, and add one flow at a time to observe the frame delivery ratio. As shown in Figure 8, when there is only one flow, most of protocols deliver frames with high successful rates, except fix transfer rate protocols like 2, 5.5, and 11 Mbps. The reason is fix transfer rate protocols can't adapt to the fast changing channel conditions, which is represented with the dynamically generated SNR. When a frame is sending with fast transfer rate over the channel with a low SNR, it would fail to demodulate the receiving frame at the recipient. As for the lowest fix transfer rate at 1 Mbps, most of channel conditions are better than the lowest SNR requirement of the rate. Thus, the frame delivery ratio would be high for the lowest fix rate transmission. Next, when flows are added into the network, more frames are generated for transmission in the same period, the broken frames caused by BER and collisions due to frame congestion gradually

depress the frame delivery ratio. Generally speaking, the rate-adaptable protocols outperform the fix rate protocols. Especially, the proposed TRAF performs the best.

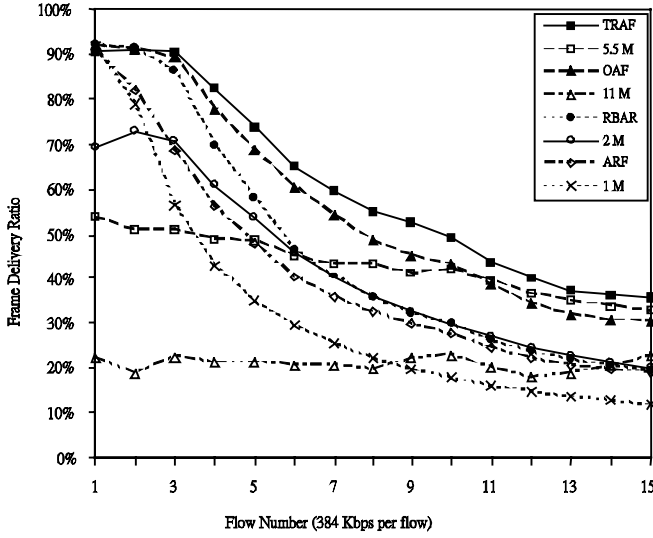


Fig. 8. Frame delivery ration vs. flows

5.3 Network Throughput

In this simulation, we would observe the network throughput of each protocol versus the network load modeled by multiple CBR flows. Here, the definition of network throughput is equivalent to that of channel utilization which is denoted as the total sent bits divided by total bandwidth at base rate. All flows keep same CBR and same frame size as previous simulation.

At the beginning with one flow as shown in Figure 9, all protocols demonstrate low network throughputs, this is due to low frame generation rate so that there is few frames ready to send. After more flows are added into the network one by one, the network throughputs are gradually increased accordingly. Where 1, 2 Mbps, ARF, and RBAR reach the saturate throughput earlier than others at about 3 to 5 flows. Other protocols keep increasing as the number of flows continuously increases. Clearly, TRAF outperform other protocols for whole simulation period. It is interesting to notice that 5.5 Mbps even has better network throughput than OAR. The reason is that OAR sends RTS/CTS control frames at base rate, and then it sends data frames with appropriate rate. Thus, RTS/CTS become essential overheads to OAR. On the contrary, 5.5 Mbps fix rate send all control frames and data frames with same high rate. Thus, the RTS/CTS overheads are shortened by higher transfer rate.

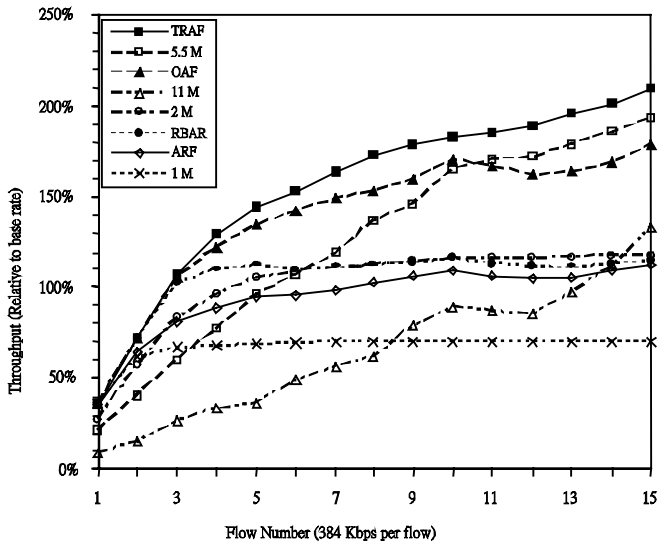


Fig. 9. Throughput vs. flows

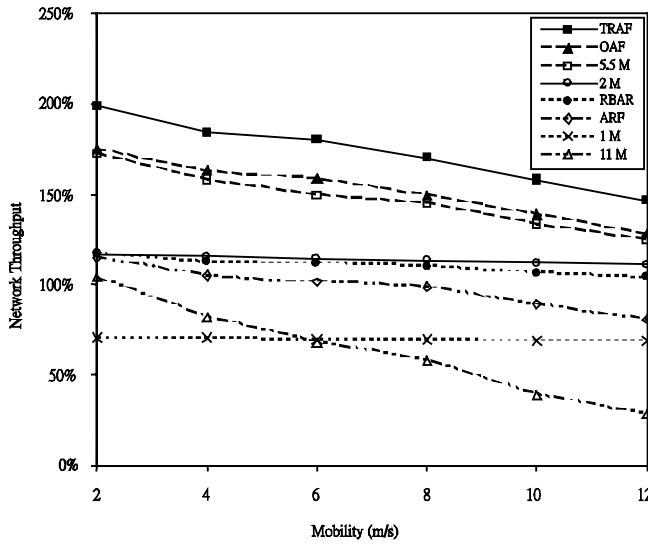


Fig. 10. Throughput vs. Mobility

In addition, as the mean value of SNR chosen for the simulation is 17, which exactly falls on the modulation scheme with transfer rate of 5.5 Mbps, this also favors the 5.5 Mbps transfer rate. Note that 1 Mbps has the lowest network throughput. Since most of time the SNR is higher than the lowest SNR requirement of 1 Mbps, and a fix

rate protocol never tries to increase its transfer rate to fully utilize good channel conditions to transfer more frames.

5.4 Mobility

Mobility affects the communicating flows in two ways. When both end nodes move closer to each other, the SNR of the flow is getting better. On the contrary, when end nodes move farther, the SNR of the flow is getting worse. We model the influences as a two-states machine, the state is stayed for a time interval generated by an exponential distribution with a mean value. Simulation was started with 10 flows with same CBR and same frame size. Simulation results are shown in Figure 10. As mobility increases, the network throughputs of most of protocols are going down accordingly, especially 11 Mbps. On the other hand, RBAR and low fix rate protocols such as 1 and 2 Mbps are insensitive to the SNR variation caused from mobility.

6 Conclusion

In this paper, we introduced TRAF rate adaptation protocol for MANET. With TRAF, a node at high channel quality can transmit multiple frames in a way of back-to-back transmissions at a rate as fast as possible. During back-to-back transmission interval, bi-directional rate exchanges are executed through the proposed option extension. With the bi-directional rate adaptation in per frame basis, TRAF enhances channel utilization with timely fast rate adaptation. Even while the recipient suffers from worse channel quality, TRAF still can adopt frame fragmentation mechanism to increase successful delivery rate and to maintain high channel throughput. Moreover, TRAF follows the 802.11 standard without any redefinition or modification. Consequently, backward compatibility can be promised as much as possible. Extensive simulation experiments are performed to compare the frame delivery ratio, network throughput, and mobility impact of TRAF to that of other protocols. The results show that significant improvements are achieved as our expectation.

References

1. "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *ANSI/IEEE Standard 802.11, Part 11*, edition 1999.
2. "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer Extension in the 2.4 GHz Band," *ANSI/IEEE Standard 802.11, Part 11*, edition 1999.
3. "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz Band," *ANSI/IEEE Standard 802.11, Part 11*, edition 1999.
4. S. Armour ; A. Doufexi, A. Nix, D. Bull, "A study of the impact of frequency selectivity on link adaptive wireless LAN systems," *Proceedings of IEEE Vehicular Technology Conference* , Vancouver, BC, Canada, Sep 24-28, 2002, v.56, n.2, pp.738-742

5. K. Balachandran, S. R. Kadaba, and S. Nanda, "Channel quality estimation and rate adaption for cellular mobile radio," *IEEE Journal on Selected Areas in Communications*, 17(7):1244–1256, July 1999.
6. G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, Mar. 2000.
7. V. Bharghavan, A. Demers, S. Shenker and Lixia Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," *Proceedings of SIGCOMM 94*, pp. 212–225.
8. R. Blake, "Wireless Communication Technology," *Delmar, Thomson Learning*, 2001.
9. F. Cali, M. Conti and E. Gregori, "IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism," *IEEE journal on selected areas in communications*, vol. 18, no. 9, September 2000.
10. B.P. Crow, I. Widjaja, J.G. Kim, and P.T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Communication Magazine*, September 1997.
11. G. Holland, N. Vaidya, P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," *Proceedings of ACM SIGMOBILE 2001*, pp.236–250, July 1, Rome, Italy
12. J.H. Gass, M.B. Pursley, H.B. Russell, R.J. Saulitis, C.S. Wilkins, and J.S. Wysocarski, "Adaptive transmission protocols for frequency-hop radio networks," *Proceedings of the 1998 IEEE Military Communications Conference*, volume 2, pages 282–286, October 1998.
13. A. Goldsmith and S. G. Chua, "Adaptive coded modulation for fading channels," *IEEE Transactions on Communications*, 46:595–602, May 1998.
14. A. Kamerman and L. Monteban, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, summer 1997, pp. 118–133.
15. P. Karn, "MACA - a new channel access method for packet radio," *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, ARRL 1990, pp. 134–140.
16. D. Qiao, S. Choi, "Goodput Enhancement of IEEE 802.11a Wireless LAN via Link Adaptation," *Proceedings of IEEE International Conference on Communications (ICC2001)*, v.7, pp. 1995–2000.
17. T. S. Rappaport, "Wireless Communications: Principles and Practice," *Prentice Hall*, 1999.
18. B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic Media Access for Multirate Ad Hoc Networks," *Proceedings of MOBICOM'02*, Atlanta, Georgia, USA, September 23–28, 2002, pp. 24–35.
19. F.A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part ii - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution," *IEEE Transactions on Communications*, COM-23(12):1417–1433, 1975.

Throughput Evaluation and Enhancement of TCP Clients in Wi-Fi Hot Spots^{*}

Raffaele Bruno, Marco Conti, and Enrico Gregori

Italian National Research Council (CNR) – IIT Institute ,
Via G. Moruzzi 1, 56126 Pisa – ITALY
{r.bruno,m.conti,e.gregori}@iit.cnr.it

Abstract. In this paper we consider a Wi-Fi *hot spot* where M users are performing TCP downloads from Internet remote servers. Our study focuses on characterizing the way the TCP flow control mechanisms affect the MAC protocol operations, and identifying the main causes of the throughput limitations shown by the TCP traffic. In particular, we show that the TCP throughput is not limited by the collision events, but by *i*) the inability of the MAC protocol to assign a higher chance of accessing the channel to the hot spot Access Point than the mobile users, and *ii*) the interaction of flow control mechanisms used at the TCP layer and the contention avoidance scheme used at the MAC layer. We propose an extension to the MAC protocol that requires only modifications of the hot spot Access Points. Our proposed enhancement allows the Access Point to send bursts of TCP packets towards the hot spot clients. We design a resource allocation protocol aimed at maximizing the success probability of the uplink transmissions by dynamically adapting the burst size to the number of users' collisions and successful transmissions. Simulations confirm the improvements of the TCP throughput achieved by our enhanced MAC protocol.

1 Introduction

Recently, the attention of manufactures and Internet providers is turning to deploying infrastructure-based wireless networks in the market of Internet public access areas, known as *hot spots*. Specifically, a hot spot can be either an area as small as a cafe and retail shop, or as large as an airport, a convention center and a hotel where people are provided with a seamless public access to the Internet. Basically, the hot spot is an area that is served by a single Wireless LAN (WLAN), or a network of WLANs where the mobile hosts access the Internet through the WLAN's Access Points (APs). Since the IEEE 802.11b technology is the dominant technology for implementing current WLANs, in this work we have considered 802.11b-based hot spots. Several researchers have devoted their efforts to investigate the performance of the 802.11 MAC protocol. Most of these

^{*} Work carried out under the financial support of the Italian Ministry for Education and Scientific Research (MIUR) in the framework of the Projects FIRB-PERF and FIRB-VICOM.

works are analytical studies ([1,2,3]), which evaluate the achievable channel utilization¹ basing on the assumption of devices operating in *saturation conditions*, i.e., transmission queues never empty. This paper highlights that this assumption doesn't fit the hot spot configuration. Specifically, the majority of applications that can be envisaged for the hot spot market, are based on TCP downloads from remote servers towards the end-users via the hot spot APs (e.g., email applications, web surfing, data retrieving and so on). Therefore, the AP is the bottleneck node that affects the performances of the whole network. Furthermore, the characterization of uplink (i.e., from user to the AP) and downlink (from the AP to the users) traffic has to be different, since the AP mostly transmit TCP data packets, whereas the hot spot clients reply back with TCP ACKs.

In this paper we focus on the analysis of the MAC protocol efficiency when the hot spot AP manages M mobile/fixed users performing TCP downloads from remote servers. Due to the complexity of the problem, in this work we have performed a simulation study, and we left to a further study the analytical characterization of the system behavior. Our study is aimed at gaining a better understanding on the causes of the severe performance limitations shown by TCP traffic in the hot spot configurations. We conducted our study considering the system from the MAC protocol perspective, and we identify the way the TCP flow control mechanisms affect the MAC protocol operations. We show that the interaction of the TCP flow control mechanisms and the MAC contention avoidance scheme impedes the hot spot clients to operate in saturation conditions. Therefore most of the optimization techniques already developed to increase the MAC efficiency are not useful in the hot spot configurations, because were derived from the saturation throughput analysis (see, e.g., the discussion and references in [3]).

We have discovered that the network contention level, expressed in terms of the average number of hot spot clients that contend for the channel bandwidth slightly increase by increasing the number of active TCP flows, as it could be expected. This observation is fundamental, because it confirms that the performance limitations are not due to the contention suffered by the multiple TCP flows, but to the inability of the MAC protocol to assign a higher chance of accessing the channel to the AP than the hot spot users. In order to overcome these limitations we propose a solution based only on the modification of the MAC protocol operations in the AP without affecting the users' behavior. Specifically, our solution allows the AP to send periodically bursts of TCP data packets towards the hot spot clients by employing a null backoff to access the channel. After sending this burst of data, the AP should wait for the users' replies. We developed a theoretical analysis to compute the burst size that the AP should adopt in order to maximize the success probability of users' transmissions. We have evaluated the proposed enhancement to the MAC protocol

¹ The channel utilization is defined as the fraction of channel bandwidth used by successfully transmitted messages. Its maximum value is referred to as protocol capacity.

via simulations. The numerical results confirm the improvements of the TCP throughput achieved by our enhanced MAC protocol.

This paper is organized as follows. In Section 2, we present the simulation results quantifying the TCP performance achievable in the considered hot spot configuration. In Section 3.1 we analytically derive the optimal AP behavior. By exploiting our analytical results, in Section 3.2 we design and evaluate our novel resource allocation protocol.

2 Study of TCP Performance in IEEE 802.11b Hot Spots

In literature measurements are already available on commercial products about the UDP [4,5] and TCP [6,7] throughput performances in 802.11 WLANs. The experimentations on test-beds are fundamental to highlight the issues of a technology, however are usually limited to observe the behavior of transport layer protocols. Furthermore, the measurements may be affected by several factors, including the link quality and the NIC's implementation details, which often preclude the possibility of conducting a rigorous study. Finally, the manufacturers don't make available to the application layer the status of relevant MAC protocol parameters, as the instantaneous backoff value, the transmission queue's occupancy, the collision events and so on. Therefore, in this paper simulations has been conducted to gather a clearer understanding of the MAC protocol operations and to identify the inefficiencies of the MAC protocol that cause the TCP performance limitations. The simulation environment we used is an extension of the one we developed in [3], which implements all the MAC and TCP protocol details. The TCP version considered is the TCP-Reno, the most worldwide adopted TCP implementation [8]. For the details on the MAC protocol overheads the reader is referred to the IEEE 802.11b specification [9].

The aims of the simulations we have conducted are: *i*) to understand the impact of multiple TCP flows on the contention level that the MAC protocol has to deal with; and *ii*) to analyze how the TCP flow control mechanisms affect the main 802.11 MAC protocol parameters, as the average backoff, the number of retransmissions, and so on. If not otherwise stated, we assume a TCP Maximum Segment Size (MSS) of 1500 bytes and a TCP advertised window size of 2^{16} bytes². The AP buffer size is assumed to be 100 MSS, that in [7] has been shown to be sufficiently large to avoid undesirable TCP unfairness. Each experiment consists of 5 simulation runs, each lasting 100 seconds of simulated time. We have considered two different network setups, which we refer to as the '*TCP case*' and the '*UDP case*', respectively. In the TCP case each STA has opened an asymptotic TCP connection with the AP. This implies that the AP has always a TCP data packet to transmit to the STAs (ftp-like traffic). In the UDP case, there are M UDP flows from the AP towards the STAs and an UDP flow from each STA towards the AP. The UDP sources are CBR flows that adopt a rate such that the transmission buffers are never empty. The UDP packets generated by the

² This implies that the TCP advertised window size is about 43 TCP packets.

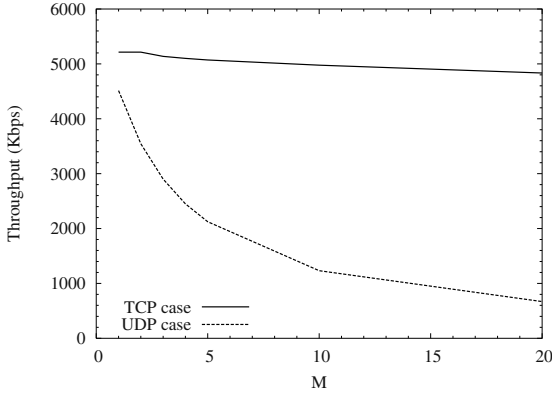


Fig. 1. The measured aggregate TCP throughput as a function of the number M of STAs

AP are 1500-bytes long packets (i.e., packets as long as the TCP data packets), and the UDP packets generated by the STAs are 40-bytes long packets (i.e., packets as long as the TCP ACK packets). The UDP case is used as *reference scenario* to quantify the impact of the TCP flow control mechanisms over the system performance. In the first set of simulations we measured the aggregate TCP throughput as a function of the hot spot population size, that is the number M of STAs. The larger is the network population, the larger is the number of devices that should contend for the channel bandwidth. According to the analysis done in other works (see, e.g., [2,1]), the larger is the number of contending devices, the larger should be the collision probability due to the CSMA/CA access scheme and thus the lower should be the channel utilization. Surprisingly, Fig. 1 shows that the aggregate TCP throughput is slightly affected by the hot spot population size, and the TCP aggregate throughput with 20 STAs is about 93% of the TCP aggregate throughput with only one STA. To better appreciate the peculiarity of the TCP case, in Fig. 1 we have also showed the aggregate UDP throughput³. It is straightforward to observe that the aggregate UDP throughput significantly decreases as the number of UDP sources increases due to the higher contention level in the network. In particular the UDP throughput obtained by the AP with 20 STAs is about 15% of the UDP throughput achieved by the AP with only one STA. Finally, it is worth pointing out that the maximum channel utilization that is obtained when there is a single TCP flow is only 0.474. Neglecting the collisions and the TCP ACK traffic we can estimate that the maximum achievable throughput (see for instance the formulas derived in [6]) is 7.28 Mbps that corresponds to a channel utilization of 0.66. Hence, the target of any optimization technique that doesn't modify the 802.11 physical layer and its overheads should be to approach this theoretical limit.

³ The aggregate TCP (UDP) throughput is defined as the sum of the throughput achieved by all the active TCP (UDP) flows.

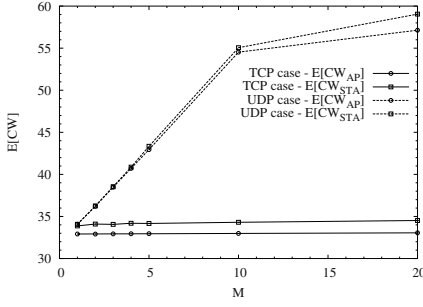


Fig. 2. Average contention window used by AP and STAs versus the number of STAs

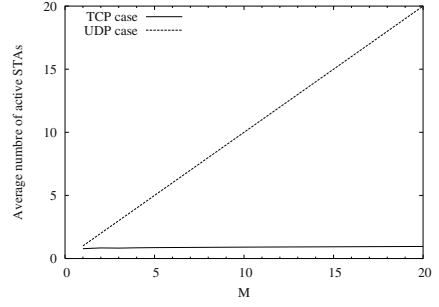


Fig. 3. Average number of active STAs after an AP's successful transmission

In the following we justify the counter-intuitive fact that the TCP throughput reduction is not due to collisions by analyzing the MAC protocol behavior. First of all, we consider the average contention windows used by the AP and the STAs, say $E[CW_{AP}]$ and $E[CW_{STA}]$ respectively. The average contention window provides a good indication of the average contention level suffered by the devices in the network. The greater is the average contention window the greater is the average time waited by the device before attempting a transmission. Fig. 2 shows the $E[CW_{AP}]$ and $E[CW_{STA}]$ values for both the TCP case and the UDP case as a function of the hot spot population. The numerical results shown in Fig. 2 highlight that in the TCP case the average contention window is slightly above 32 slots, independently of the M value, indicating that both the AP and STAs experiences a few collisions. On the other hand, for the UDP case the average contention window increases up to 59 when there are 20 STAs. This means that the devices have a significant probability to suffer at least a collision and to double the contention window using the 64-slots value. The results on the average contention windows clearly explain the difference about the throughput obtained in the TCP case and in the UDP case: TCP flows suffer a low number of collisions and there is a negligible impact of the number of TCP flows on the collision probability. This is an essential point to understand the TCP performances and in the following we provide further results to motivate this behavior.

In Fig. 3 we show the average number of STAs that after an AP's successful transmission have a packet to transmit. For the UDP case it is straightforward to observe that the number of STAs with a packet to transmit is always equal to M . On the other hand, the STAs' activity in the TCP case is strongly affected by the TCP flow control mechanisms since the amount of TCP acknowledgment traffic the STAs have to reply back to the AP depends on the amount of the TCP data traffic the AP succeeds in delivering to the STAs. Specifically, the STAs are the TCP receivers, hence they can have a new TCP ACK to transmit to the AP only after the reception of a TCP data packet from the AP. The TCP

ACK generation process is further complicated by the “Delayed ACK” technique that causes the ACK generation to be delayed for a short period of time [8]. The TCP standard also recommends that an ACK should not be delayed for more than two data packets. The TCP specification mandates that the delay must be less than 0.5 seconds, but most of the implementations use a 200 ms delay [8].

In conclusion we can state that, although the AP has always data packets to transmit, one or more STAs can be inactive, i.e., have empty transmission queues, because they have to wait to receive TCP data packets before having TCP ACKs to reply back. This behavior motivates why the average number of STAs that are active and contend for the channel bandwidth with the AP is significantly lower than M . A further relevant outcome can be driven from the results shown in Fig. 3. Specifically, the average number of active STAs, that is a measure of the average contention level in the network, slightly increases by increasing the hot spot population size, passing from 0.78 for $M = 1$ to 0.95 for $M = 20$. The explanation of this phenomenon can be found in the interaction of the TCP flow control mechanisms and the MAC contention avoidance scheme. The more traffic the AP sends to the STAs the more STAs become active. In addition, the larger is the number of active STAs the lower is the probability that the AP can experience a successful transmission. Thus the STAs tend to empty their transmissions queues and to become inactive. This interaction between the traffic sent by the AP and the traffic replied back to the AP by the STAs operates as an intrinsic closed-loop control that stabilizes the network, limiting the contention level to a few STAs on average. The results shown in Fig. 3 provide an explanation also for the fact that $E[CW_{AP}]$ is always lower than the $E[CW_{STA}]$ as indicated in Fig. 2. Specifically, it is possible that the AP transmits its packets without other STAs contending for the channel, thus avoiding collisions. On the other hand, the STAs have always at least the AP contending for the channel resources.

3 Enhancing Hot Spot Throughput

In this section we propose a simple enhancement of the 802.11 MAC protocol in order to improve the TCP throughput in the hot spot configurations considered in this work. We can identify two main concerns when proposing modifications of the MAC layer. The first one is that the modified MAC protocol could require hardware upgrades to be implemented, which is infeasible given the wide deployment of standard IEEE 802.11 NICs. Nevertheless, it is not infeasible to propose extensions to the MAC layer that involve only firmware upgrades in the network cards of the hot spot APs, without any modification in the network cards installed in the mobile hosts. The second concern is related to the compatibility requirements that any protocol extension should fulfill. It is desirable that the protocol modifications are designed in such a way that the behavior of the standard protocol is not hampered by the operations of the enhanced one. In other words, mixed scenarios should be supported where standard and modified network cards can safely inter-operate without causing performance

degradations to the users owing the network cards implementing the standard protocol. The solution we propose takes into account both of these concerns as we will explain in the following.

As briefly described in Section 1, a considerable research activity has been focused to increase the MAC protocol efficiency in terms of the maximum achievable channel utilization. This goal was mainly obtained by modifying the back-off procedure in such a way to minimize the collision probability ([10,3] and references herein). A wise choice of the contention window, which should be dynamically tuned according to the network configuration (i.e., the number of stations in the network) and traffic conditions (i.e., the distribution of the message lengths), can lead to attain significant improvements as far as the protocol capacity. However, these policies are not effective in the hot spot configurations where the causes of throughput reduction are not the collisions events, but the useless overheads that precede TCP data transmissions. Our solution is to let the AP to make its transmission attempts by using a null backoff value. This implies that the AP can start a new transmission attempt immediately after it senses the channel to be idle for a DIFS interval. This choice as a twofold remarkable result: the time required to successfully transmit a TCP packet is reduced, and the probability that an AP transmission collides with concurrent STAs' transmissions is negligible. To force the AP to use a null backoff is easy to implement because, although the binary truncated exponential backoff algorithm distribution is usually hardwired in the NIC, the distribution parameters can be set in the NIC driver. Thus, to implement a null backoff it is sufficient to set to zero the maximum contention window value in the AP's NICs. From the STAs' perspective, the 802.11 MAC protocol holds its correctness because they can continue to operate with a standard backoff.

It is worth pointing out that the optional access scheme proposed by the IEEE 802.11 standard, the *Point Coordination Function (PCF)* [9], is also based on the use of AP's transmissions with higher priority than STAs' transmissions. However, significant differences can be identified between our approach and the PCF. In the PCF each AP's transmission is followed by the STA's reply. When the STA has no traffic to send either to the AP or to another STA, it is mandated to deliver a null packet, further reducing the protocol efficiency. Generally, assigning a higher priority to the AP's transmissions by using a null backoff is not a sufficient condition to increase the throughput. As it will be explained in the following, differently from the PCF, we propose to separate the time intervals where the AP is allowed to deliver its traffic, from the time intervals where the STAs deliver their traffic. The duration of these time intervals will be dynamically selected in such a way to maximize the rate of STAs' successful transmissions. Basically the PCF is a polling schemes where the AP decides the order the STAs are allowed to send packets: STAs that are not polled are blocked by the AP. In our scheme the AP sends its burst of data packets almost in a contention-free manner, but it doesn't control the STAs transmissions. In fact, during the time interval reserved to the STAs' transmission, the STAs will regulate the channel access according to the standard DCF contention-based scheme.

According to our protocol, when the AP decides to perform a new transmission it seizes the channel and sends a burst of l TCP data packets. Taking into account the Delayed ACK mechanism, these l transmissions can cause at most the generation of $\lfloor l/2 \rfloor$ new TCP ACKs in the STAs. After the AP's delivery of its burst of data, the AP must let to the STAs the opportunity to transmit their queued TCP ACKs before the AP starts a new burst of transmissions. Let assume that there are m active STAs in the network, i.e., STAs with at least a TCP ACK to transmit, and that all the m STAs are using the minimum contention window, say w (Fig. 2 indicates that this assumption is a good approximation in our scenario). This assumption implies that in the next w virtual slots some of the m stations will surely perform a transmission attempt, since each STAs will uniformly select a backoff in the range $[0, \dots, w-1]$. We use the same notation followed by Bianchi in [2] where the virtual slots can be: *i*) empty slots with duration t_{slot} , when no stations are transmitting; *ii*) "collision" slots with duration T_c when two or more STAs collide; and *iii*) "successful" slots with duration T_s , when a single STA is transmitting. Therefore the virtual slots haven't the same *weight*. An optimal choice for the l value is the value that activates a number m of STAs such that the channel utilization during the next w virtual slots (when the STAs' transmissions are allowed) is maximal. We define as *success rate* the ratio between the number of successful slots and the time occupied by the w virtual slots, that is

$$\text{success rate} = \frac{N_s}{N_i \cdot t_{slot} + N_s \cdot T_s + N_c \cdot T_c}, \quad (1)$$

where N_i , N_s and N_c are the number of idle slots, successes and collisions during the w virtual slots. In order to estimate the optimal m , say m^* , we need to derive a relationship between the number of the active STAs and the success rate, such that it could be maximized. In the following section we develop an analytical framework that allows us to calculate the m^* value.

3.1 Maximizing the Success Rate

The problem we address in this section is to determine the number m of contending STAs that should be active after the AP has sent its burst of TCP data packets, in order to maximize the success rate in a window of w virtual slots⁴. To achieve our goal we need to calculate how many of these w virtual slots will be idle slots, how many will be collision slots and how many will be successful slots. Henceforth, given that there are m active STAs, we indicate the number of successful slots that will be observed during a window of w virtual slots as $E[N_s]_m^w$, the number of collisions as $E[N_c]_m^w$, and the number of idle slots as $E[N_i]_m^w$. First of all we need to express the probability, given m active STAs, that a virtual slot is a success, say $P_s(w, m)$, a collision that involves k STAs, say $P_c(w, m, k)$, or an idle slot, say $P_i(w, m)$. Following the approach used in [1],

⁴ The standard MAC protocol uses an initial contention window of 32 slots, but we have carried out an analysis that is valid for a general contention window.

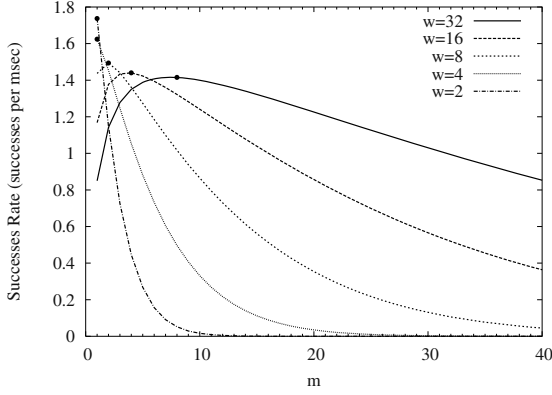


Fig. 4. Success rate as a function of m for different contention window size w

and indicating with p_w the probability that a STA is transmitting in a slot conditioned to the fact that it will try to access the channel within the following w virtual slots, we can write

$$P_s(w, m) = m \cdot p_w \cdot (1 - p_w)^m, \quad (2a)$$

$$P_c(w, m, k) = \binom{m}{k} p_w^k \cdot (1 - p_w)^{m-k}, \quad (2b)$$

$$P_i(w, m) = (1 - p_w)^m. \quad (2c)$$

To derive p_w it is enough to observe that it is equiprobable that each STA tries to transmit in any of the following w virtual slots, therefore $p_w = 1/w$. By exploiting formulas (2), the following Lemma defines recursive algorithms to derive $E[N_s]_m^w$, $E[N_c]_m^w$ and $E[N_i]_m^w$.

Lemma 1. *If m active STAs uniformly try a transmission attempt during w consecutive virtual slots, the number of successful, collision and idle slots during these w virtual slots is:*

$$E[N_s]_m^w = P_s(w, m) \{1 + E[N_s]_{m-1}^{w-1}\} + \sum_{k=2}^m P_c(w, m, k) E[N_s]_{m-k}^{w-1} + P_i(w, m) E[N_s]_m^{w-1}, \quad (3a)$$

$$E[N_c]_m^w = P_s(w, m) E[N_c]_{m-1}^{w-1} + \sum_{k=2}^m P_c(w, m, k) \{1 + E[N_c]_{m-k}^{w-1}\} + P_i(w, m) E[N_c]_m^{w-1}, \quad (3b)$$

$$E[N_i]_m^w = P_s(w, m) E[N_i]_{m-1}^{w-1} + \sum_{k=2}^m P_c(w, m, k) E[N_i]_{m-k}^{w-1} + P_i(w, m) \cdot \{1 + E[N_i]_m^{w-1}\}, \quad (3c)$$

Proof. Omitted due to the space constraints, the reader is reminded to [11].

Lemma 1 can be used to calculate the success rate as a function of the m value, hence determining the m value that maximizes it. Unfortunately, the recursive algorithm requires a considerable computational cost as m increases. To solve this problem we have also developed an efficient iterative procedure to compute formulas (3). This procedure is based on the construction of three matrixes with $(w+1)$ rows and $(m+1)$ columns, $\mathbf{S} = \{s_{i,j}\}$, $\mathbf{C} = \{c_{i,j}\}$ and $\mathbf{I} = \{i_{i,j}\}$, whose elements are defined, respectively, as: $s_{i,j} = E[N_s]_j^i$, $c_{i,j} = E[N_c]_j^i$ and $i_{i,j} = E[N_i]_j^i$, for $i=0, 1, 2, \dots, w$ and $j=0, 1, 2, \dots, m$. The matrices' elements are evaluated by exploiting the formulas derived in Lemma 1. For instance, the $s_{i,j}$ is

$$s_{i,j} = P_s(i, j) \cdot \{1 + s_{i-1,j-1}\} + \sum_{k=2}^m P_c(i, j, k) \cdot s_{i-1,j-k} + P_i(i, j) \cdot s_{i-1,j} \quad (4)$$

Hence, the quantities defined in formulas (3) are the last element on the matrixes' diagonals. Furthermore, once we have calculate $E[N_s]_m^w$, we have for free the $E[N_s]_j^w$ for $j \leq m$, that are given by the last row of \mathbf{S} . Clearly, the same holds for \mathbf{C} and \mathbf{I} . By observing (4) it is straightforward to notice that the $s_{i,j}$ element depends only on the element of the first j columns of the previous row. Therefore, to apply the iterative procedure we need only to know *a priori* the elements $\{s_{1,j}, s_{i,0}\}$, $\{c_{1,j}, c_{i,0}\}$ and $\{i_{1,j}, i_{i,0}\}$ for $i = 0, 1, 2, \dots, w$ and $j = 0, 1, 2, \dots, m$ ⁵. Let us start from the first couple. The index i indicates the size of the window where all the j STAs will try a transmission attempt. Hence, $i = 1$ implies that all the j STAs will access the channel, thus we can count a successful slot only for $j = 1$. On the other hand if $j = 0$ we cannot have transmissions. To summarize

$$s_{i,0} = 0 \quad \text{for } i = 0, 1, 2, \dots, n \quad , \quad s_{1,j} = \begin{cases} 1 & \text{if } j = 1 \\ 0 & \text{if } j = 2, \dots, m \end{cases} \quad (5)$$

In the case of collisions the reasoning is clearly the opposite. In fact, if $i = 1$ we have to count a collision for $j > 1$. Hence

$$c_{i,0} = 0 \quad \text{for } i = 0, 1, 2, \dots, w \quad , \quad c_{1,j} = \begin{cases} 0 & \text{if } j = 1 \\ 1 & \text{if } j = 2, \dots, m \end{cases} \quad (6)$$

The case of idle slots is different. If $i = 1$ and $j > 0$, there will be at least a transmission attempt in that slot, therefore we cannot count idle slots. If $j = 0$ we cannot have transmissions, and all the remaining i virtual slots will be idle slots. Hence

$$i_{i,0} = i \quad \text{for } i = 0, 1, 2, \dots, w \quad , \quad i_{1,j} = 0 \quad \text{for } j = 1, 2, \dots, n \quad (7)$$

Using the initial conditions derived in formula (5), (6) and (7), we are finally able to compute the quantities defined in Lemma 1. To evaluate the duration of successful slots and collisions slots we have considered STAs sending 40-bytes

⁵ It is straightforward to note that $s_{0,j} = c_{0,j} = i_{0,j} = 0$ for $j = 0, 1, 2, \dots, m$.

long packets⁶ and introduced all the MAC protocol overheads. Fig. 4 shows the success rate and the plotted curves can be exploited to easily derive the m^* value. The numerical results indicate that for all the w values analyzed the success rate is maximized for a number m^* of STAs such that $m^* = w/4$. This is a not-intuitive condition, and it was identified by using our analytical study. Further studies of this nice property are an ongoing activity beyond the scope of this paper. It is worth pointing out that this property depends on the specific setting of the MAC protocol overheads. Modifying the interframe spaces, will cause the change of the m^* value. We can observe that the m^* is lower than the m value that simply maximizes the number of successes during a window w of virtual slots. This can be explained by noting that to maximize the success rate, we try to maximize the number of successes per unit time, hence taking into account also the high cost due to collision overheads.

To summarize, if the AP operates in such a way that, after sending a burst of l TCP data packets, it has activated not more than m^* STAs, then the AP maximizes the STAs' success rate in the following contention window. To compute the m^* that maximizes the success rate we have assumed that all the STAs are using the minimum contention window w , as we have assumed that they have the same chance to try a transmission attempt during a window of w virtual time slots. However, even if the number of collision slots observed when using m^* is low (for $w = 32$ we have on average 0.77 collision slots when $m = m^*$), it cannot be neglected. Specifically, after the AP has sent a new burst of TCP data packets, in the network we will have the “new” STAs that has been activated by the new TCP packets, but also the “old” STAs that either suffered a collision in the previous contention window or didn't access the channel. To conclude, the AP has to behave in such a way that after sending a burst of l TCP data packets, in the network there are not more than m^* active STAs, but counting both newly activated STAs and previously activated STAs that didn't experience a successful transmission attempt in the previous contention window. In the following section we design and evaluate a resource allocation protocol that, exploiting our analytical results, allows the AP to increase the aggregate TCP throughput. This goal is achieved by dynamically adapting the burst size l of TCP data packets the AP sends according to the number of STAs' collisions and successful transmissions observed on the channel during a window of w virtual slots .

3.2 A Dynamic Resource Allocation Protocol for the Hot Spot AP

As said in Section 3, we propose that the AP performs its transmission attempts by using a null backoff value. This implies that the AP can start a new transmission attempt immediately after it senses the channel to be idle for a DIFS interval. The AP will send a burst of l TCP data packets, and then it will wait for the STAs replying back their TCP ACKs. How long does the AP have to wait for? The time needed to observe on the channel a number of idle slots, STAs'

⁶ This is the typical size of the TCP ACK.

successful transmissions and collisions than sum up to w (where $w = 32$ to be compliant with the standard minimum contention window). Therefore the AP's behavior is cyclic: the AP's bursts of data are interleaved by 32 virtual time slots in which it doesn't participate to the channel contention. We indicate the i^{th} AP's cycle as γ_i . We show in Fig. 5 the structure of the i^{th} AP's cycle. As explained in Section 2, due to the Delayed ACK mechanism used by the TCP protocol, at most $\lfloor l/2 \rfloor$ TCP ACK can be generated in the STAs if the AP sends consecutively l TCP data packets. By assuming that before the AP burst delivery, the STAs have empty transmission queues, the number m of active STAs can be at most $\lfloor l/2 \rfloor$ ⁷. This assumption provides an approximation of the number of TCP ACKs in the system during an AP's cycle. The relationship between the number of TCP packets the AP has sent, and the number of STAs that have TCP ACKs to reply back is clearly more complex than the one we adopt in the following discussion. The development of a more precise characterization of the STAs' activity is an ongoing activity beyond the scope of this work. In fact, in this paper we aim at proving the effectiveness of our approach rather than to derive the optimal policy. Hence we will show that is feasible to increase the aggregate TCP throughput of the hot spot clients basing on the reduction of protocol overheads and the maximization of the uplink success rate.

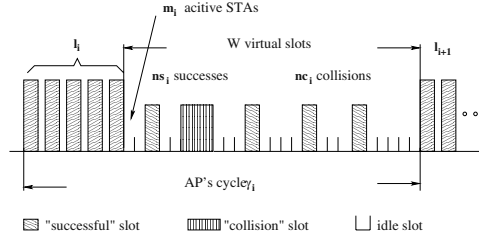


Fig. 5. Structure of channel events between during the AP's cycle γ_i .

According both to the analysis done in Section 3.1 and to the previous assumptions on the relationship between m and l , we should select the l value in such a way that $\lfloor l/2 \rfloor = m^*$ to maximize the STAs' success rate⁸. However, this choice can overload the network because the resulting scheme doesn't take into account that, due to collisions, STAs that were trying to access the channel in the γ_i AP's cycle, could contend also in the γ_{i+1} AP's cycle. Let us indicate as ns_i the number of STAs' successful transmissions occurred during γ_i , and as nc_i the number of STAs' collisions occurred during γ_i . To properly evaluate the

⁷ The number of active STAs after a burst of l TCP data packets will be exactly $\lfloor l/2 \rfloor$ only if we also assume that the AP doesn't send more than two TCP packets to the same STA.

⁸ It is worth reminding that the number m^* of STAs that should be active to maximize the success rate depends only on the w value and the packet size, but it is not affected by the hot spot population size.

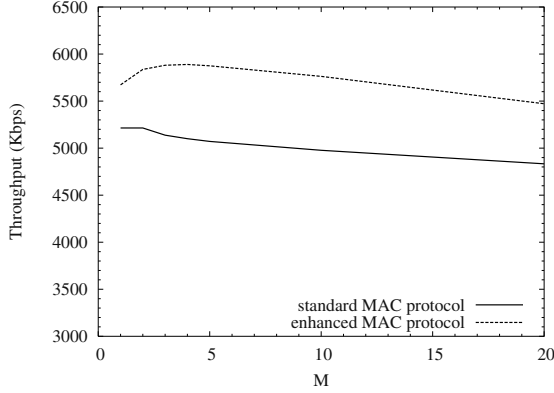


Fig. 6. The measured aggregate TCP throughput as a function of the number M of STAs

number of TCP data packets the AP should send in the $(i+1)^{th}$ burst of transmissions, say l_{i+1} , we introduce a tuning factor that takes into account both the ns_i and nc_i values. The aim of this tuning is to assure that the number m_{i+1} of STAs that will try to access the channel during the γ_{i+1} is around the optimal m^* value. Taking into account all these aspects, the AP should select the burst size l_{i+1} in the following way

$$\begin{cases} l_{i+1} = 2 \cdot \{m^* - (\lfloor l_i/2 \rfloor - ns_i) - nc_i\} & \text{if } ns_i < \lfloor l_i/2 \rfloor \\ l_{i+1} = 2 \cdot \{m^* - nc_i\} & \text{otherwise,} \end{cases} \quad (8)$$

In formula (8) we differentiate the cases when the STAs perform a number of successful transmissions that is lower than the estimated number of active stations (i.e., $ns_i < \lfloor l_i/2 \rfloor$), or not. In the first case, the optimal m value is decremented not only by the number of the observed collisions during γ_i , but also by the number of estimated STAs that have not yet transmitted their TCP ACKs.

In the following we show the numerical results obtained through simulations of an hot spot whose AP behaves accordingly to the strategy detailed in (8). In particular, we consider the same scenario used in Section 2 in order to verify the improvement achieved by adopting our resource allocation protocol. In Fig. 6 we show the aggregate TCP throughput that is achieved by the hot spot users when the AP implements either the 802.11 MAC standard protocol or employs our enhanced resource allocation protocol. We can observe that the improvement in the TCP throughput can be up to the 15%. The maximum aggregated TCP throughput we measured was 5.9 Mbps, which corresponds to a channel utilization of 0.54. As shown in Section 2, the theoretical limit obtained by ignoring collisions and TCP acknowledgments is about 0.66. Hence, a margin is still left to further improvements. It is worth pointing out that the modifications we proposed to the MAC protocol operations in the AP don't need any explicit

interaction with the upper layer protocols⁹. Although the number of hot spot clients is information easily obtained by counting the number of IP addresses the AP has assigned, knowing the exact number of TCP flows that are sending traffic could be difficult. However, since the m^* value depends only on the CW_{MIN} parameter, we don't need to estimate the number of current active TCP flows. The shown results confirm that prioritizing the AP's transmissions and tuning the number of AP's transmissions to the STAs' activity level can significantly increase the aggregated TCP throughput. The design of further resource allocation policies based on more precise characterization of the STAs' activity is an ongoing research activity.

References

1. Calí, F., Conti, M., Gregori, E.: Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. **8** (2000) 785–799
2. Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function. **18** (2000) 1787–1800
3. Bruno, R., Conti, M., Gregori, E.: Optimization of Efficiency and Energy Consumption in p -Persistent CSMA-Based Wireless LANs. *IEEE Trans. Mob. Comp.* **1** (2002) 10–31
4. Arranz, M.G., Agüero, R., Muñoz, L., Mahönen, P.: Behavior of UDP-Based Applications over IEEE 802.11 Wireless Networks. In: *Proceedings of PIMRC'01*, San Diego, CA (2001) 72–77
5. Xylomenos, G., Polyzos, G.C.: TCP and UDP Performance over a Wireless LAN. In: *Proceedings of IEEE Infocom 1999*, New York, NY (1999) 439–446
6. Pilosof, S., Ramjee, R., Raz, D., Shavitt, Y., Sinha, P.: Understanding TCP fairness over Wireless LAN. In: *Proceedings of IEEE Infocom 2003*, San Francisco, CA (2003)
7. Heusse, M., Rousseau, F., Berger-Sabbatel, G., Duda, A.: Performance Anomaly of 802.11b. In: *Proceedings of IEEE Infocom 2003*, San Francisco, CA (2003)
8. Stevens, W.: *TCP Illustrated, Volume 1: The Protocols*. Addison-Wesley, New York, NY (2001)
9. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification/Amendment 2: Higher-speed Physical Layer (PHY) in the 2.4 GHz band. ANSI/IEEE, Piscataway, NJ (2001)
10. Bononi, L., Conti, M., Donatiello, L.: Design and Performance Evaluation of a Distributed Contention Control (DCC) Mechanism for IEEE 802.11 Wireless Local Area Networks. *Journal on Parallel and Distributed Computing* **60** (2000)
11. Bruno, R., Conti, M., Gregori, E.: Throughput Evaluation and Enhancement of TCP Clients in Wi-Fi Hot Spots. Technical report, IIT – CNR Pisa (2003)

⁹ More complex scenarios with a mix of TCP and UDP flows will require a sort of explicit interaction with the upper layers, but the investigation of this case is left to future activities.

An Adaptive IEEE 802.11 MAC in Multihop Wireless Ad Hoc Networks Considering Large Interference Range*

Tzu-Chieh Tsai and Chien-Ming Tu

Department of Computer Science
National Chengchi University
{ttsai, g8912}@cs.nccu.edu.tw

Abstract. The IEEE 802.11 standard is the most popular Medium Access Control (MAC) protocol for wireless local area networks. However, in multihop wireless ad hoc networks, the IEEE 802.11 MAC protocol will suffer from more serious hidden terminal and exposed terminal problems than those in single hop WLANs. More specifically, it is due to the “large” interference range and the “large” carrier sensing range. In this paper, we focus on the collisions caused by the existence of large interference range in multihop wireless ad hoc networks and propose an adaptive IEEE 802.11 MAC (AMAC) that makes two simple modifications of IEEE 802.11 RTS/CTS handshake to dynamically adjust the transmission and reception according to the shared medium status near transmitter and receiver, respectively. Simulation results show that our method can lessen interferences and increase system throughput as compared with IEEE 802.11 MAC in the multihop wireless ad hoc networks.

1 Introduction

People have been dreaming to communicate with anyone, anytime, anywhere. With the recent advances in wireless technologies and the development of mobile computing devices, it is now possible and popular to build high-speed wireless systems that are easy to install and operate. Wireless is the only medium that can facilitate such communications.

A wireless ad hoc network is a collection of mobile nodes equipped with wireless transceivers that form an autonomous network without the help of any fixed networking infrastructure. A node can transmit data packets directly to other nodes which are within its radio coverage range or via multihop store-and-forward relay to nodes outside the range. Such network received considerable attention in recent years in both commercial and military applications due to its attractive properties of building a network on the fly without requiring any pre-planned infrastructure such as base stations or a central controller. These factors make the study of multihop ad hoc networks very interesting. This paper will focus on issues of multihop networks.

* This work was supported under NTPO project NSC91-2219-004-004

Because the wireless is a broadcast medium, it is inevitable that multiple devices access medium at the same time, thus results in garbled data so-called collision. And, the frequency is a scarce resource and is a shared medium. Efficiently controlled access of this shared media becomes a complicated and important task. Therefore, many medium access control (MAC) protocols were developed [1][2]. Among all these standards, e.g. IEEE 802.11, HIPERLAN1/2, Bluetooth; the IEEE 802.11 is the most popular protocol used in both WLANs and MANETs (Mobile Ad-hoc Networks).

In this paper, we study interference problems resulted from large interference range when the IEEE 802.11 MAC is used in multihop wireless ad hoc networks and then propose two modifications of the RTS/CTS handshake in the IEEE 802.11 MAC. The large interference range means that the range of interference is larger than the transmission range. In the past, most researches assumed that hidden terminals are located inside the transmission range, few of them considered the large interference range, which is very serious in multihop wireless ad hoc networks.

The rest of this paper is organized as follows. In Section 2 we review the IEEE 802.11 MAC protocol, hidden terminal and exposed terminal problems. In Section 3 we explain the large interference range and its influences. Then we present our proposed modifications in detail including transmitter side control and receiver side control in Section 4. Simulation results using NCTUns network simulator are given in Section 5 and we then conclude this paper in Section 6.

2 Background

The IEEE 802.11 MAC protocol [3][4] defines two different access method, one is the fundamental mechanism to access the medium called distributed coordination function (DCF), another is an optional point coordination function (PCF), is a centralized MAC protocol able to support collision free and time bounded services. Because the PCF is not suitable for distributed ad hoc networks, we now describe the DCF in detail.

2.1 The DCF of the IEEE 802.11 MAC Protocol

The DCF is also known as carrier sense multiple access with collision avoidance (CSMA/CA). For a mobile node to transmit, it shall sense the medium to determine if another mobile node is transmitting. If the medium is not determined to be busy for greater than or equal to a DIFS (DCF IFS) period, the transmission may proceed. If the medium is determined to be busy, the mobile node shall defer until the end of the current transmission. After deferral, or prior to attempting to transmit again immediately after a successful transmission, the mobile node shall select a random backoff interval and shall decrement the backoff interval counter while the medium is idle. Whenever the backoff timer reaches zero, transmission shall commence.

2.2 The Hidden Terminal and Exposed Terminal Problems

The CSMA/CA mechanism was designed to avoid collisions, however, resulted in the hidden terminal problem [5][6], and exposed terminal problem [6].

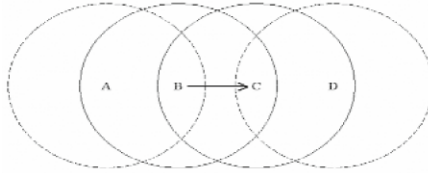


Fig. 1. Hidden terminal and exposed terminal.

A hidden terminal is one that is within the range of the intended receiver but out of range of transmitter. Consider the case shown in Figure 1. Station B is transmitting to station C. Station D cannot hear the transmission from B. During this transmission when D senses the channel, it falsely thinks that the channel is idle. If station D starts a transmission, it interferes with the data reception at C. In this case station D is a hidden terminal to station B. Hence, hidden terminals can cause collisions on data transmission.

Exposed terminals are complementary to hidden terminals. An exposed terminal is one that is within the range of the transmitter but out of range of the receiver. In Figure 1, station A can hear the transmission from B. When station A senses the channel, it thinks that the channel is busy. However, any transmission by station A does not reach C, and hence does not interfere with data reception at station C. In this case, station A is an exposed terminal to station B. Ideally, station A can send simultaneously to other receivers without interfering C's reception. Therefore, if exposed terminals are prohibited to transmit, it will underutilize available bandwidth.

2.3 The RTS/CTS Handshake in IEEE 802.11 MAC

The main task of a MAC is to avoid collisions. In order to eliminate hidden terminal problems, the IEEE 802.11 MAC protocol defines an optional four-way handshaking technique, known as request-to-send/clear-to-send (RTS/CTS) mechanism. A mobile node that wants to transmit, follows rules explained in Section 2.1, and then, instead of the data frame, preliminarily transmits a special short frame called request to send (RTS) to reserve the channel. When the destined node receive the RTS frame, it responds, after a SIFS, with a clear to send (CTS) frame. The transmitting node is allowed to transmit its frame only if the CTS frame is correctly received. All other nodes that hear either the RTS and/or the CTS set their virtual CS indicator, called a network allocation vector (NAV), for the given duration indicated in the RTS/CTS frame. The NAV state is combined with physical carrier sense function to indicate the busy state of the medium. This mechanism reduces the probability of the receiver side collision caused by a node that is hidden from the transmitter. Therefore, all other

nodes inside the transmission range of transmitter/receiver that hear the RTS/CTS will defer their transmission and thus avoid collisions caused by hidden terminals.

3 Large Interference Range

In recent years, more and more researchers have realized large interference range [7][8][9]. The large interference range means that the range of interference is larger than the transmission range as following.

3.1 The Large Interference Range

Considering the signal propagation, some nodes that are out of the transmission range of both the transmitter and the receiver, may still interfere with the receiver. This situation happened rarely in the singlehop WLAN because almost all the mobile nodes are within each other's transmission range. But in the multihop wireless ad hoc networks, the phenomenon does exist and becomes a serious problem. To prove this fact, [7] uses a simple analytic model to show that in the open space environment, the interference range of a receiver is 1.78 times the transmitter-receiver distance as Figure 2 shows. That is, if the distance between the transmitter, S, and receiver, R, is d , then the interference range of receiver is $1.78*d$, which may be larger than the transmission range. Any stations inside this interference range once transmitting to other nodes can interfere R's reception. This result overthrows the early assumption of interfering nodes are within the transmission range.

3.2 The Large Interference Range in the NS-2 Simulator

There is also another thing needed to be noticed that the carrier sense wireless networks are engineered in such a way that the carrier sensing and interference range is typically larger than the range at which receivers are willing to accept a packet from that same transmitter [10]. Many researchers use the famous NS-2 network simulator from Lawrence Berkeley National Laboratory (LBNL) [11] with extensions from the MONARCH project at Carnegie Mellon [12] to estimate their proposals. These extensions include a set of mobile ad hoc network routing protocols and an implementation of BSD's ARP protocol, as well as an IEEE 802.11 MAC protocol.

In the NS-2 network simulator, the interfering range (and sensing range) is larger than the communication range. It is implemented using a simple BER model: if the received power level of the incoming frame below the carrier sense threshold, the frame is discarded as noise; if the received power level is above the carrier sense threshold but below the receive threshold, the frame is passed to the MAC layer but marked as a packet in error; otherwise, the frame is simply forwarded up to the MAC layer.

In the IEEE 802.11 MAC of NS-2, when a mobile node is receiving a frame, and some other frame is also transmitted during its reception, two cases could happen. If the power level of the packet already being received is at least 10 dB greater than the received power level of the new packet, the MAC layer assumes “capture”, discards the new packet, and allows the receiving interface to continue with its current receive operation; otherwise, a collision occurs and both frames are dropped. Here the 10 dB is a typical value in network simulator while modeling the receiver's capability to capture signal from noise. As shown in Figure 3, a collision happens when a signal propagated inside carrier sense range arrives earlier than a signal propagated inside a transmission range arrives. Because the later stronger signal collides the formal weak signal receiving. This is also an instance of large interference range.

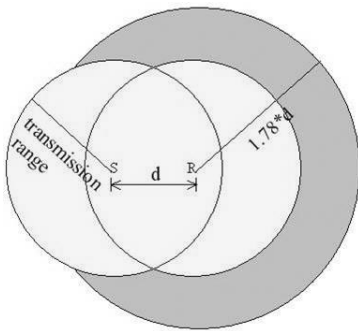


Fig. 2. Large interference range

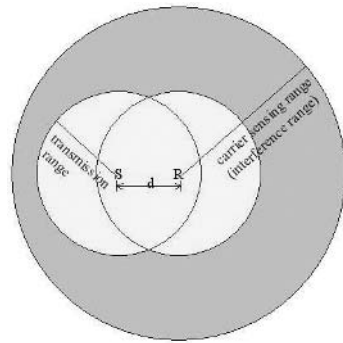


Fig. 3. Large interferences range in NS-2

3.3 The Influences of Large Interference Range

The RTS/CTS handshake was proposed to solve the hidden terminal problem based on an assumption that hidden nodes are within transmission range of the receiver. With overhearing the CTS control frame, the hidden terminals near the receiver can be inhibited so that the RTS/CTS can eliminate most of interference.

Ideally, the RTS/CTS handshake can eliminate most of interference. However, if the large interference range is concerned, the RTS/CTS is not so effective. Nodes located outside the transmission range of both transmitter and receiver may still interrupt reception [7], even if the RTS/CTS is supported. Because nodes located outside transmission range of transmitter and/or receiver cannot receive the RTS/CTS correctly, they won't keep silent if they happen to have something to transmit. Moreover, the large interference range will cause more collisions either on control frames or data frames. This results in more serious problems such as TCP instability and unfairness [8]. This situation is infrequent in an 802.11 basic service set, because all nodes can sense each other's transmissions. However, in an ad hoc network, it becomes a serious problem due to the large distribution of mobile nodes and the multihop operation.

In [7], they propose a simple MAC layer scheme called Conservative CTS Reply (CCR). An intended receiver only replies a CTS frame to a RTS initiator when the receiving power of that RTS frame is larger than a certain threshold (CTS_REPLY_THRES-HOLD). This CTS_REPLY_THRES-HOLD should be larger than the threshold required for a node to successfully receive a packet. For example, let R_{ix} denotes the transmission range. The value is chosen as a receiving power at a receiver which is $0.56 \cdot R_{ix}$ away from the transmitter. Since when the transmitter-receiver distance is smaller than $0.56 \cdot R_{ix}$, the interference range is smaller than $1.78 \cdot (0.56 R_{ix})$, which is R_{ix} . So the whole interference area is covered by RTS/CTS handshake. Therefore it can totally eliminate the collisions caused by large interference range. The illustration of transmission range, interference range, and CTS reply range are shown in Figure 4, where the CTS reply range is the range of a receiver willing to reply CTS back to the RTS initiator.

However, the CCR is over conservative that limits the available radio utilization range. Only transmitters inside $0.56 \cdot R_{ix}$ of receiver are allowed to transmit, and the available transmission area is reduced to 31.36% (i.e. 0.56^2) of physical transmission area, even if there are no hidden terminals in the interference range. Moreover, it is not effective if mobile nodes are sparse or the traffic is not heavy.

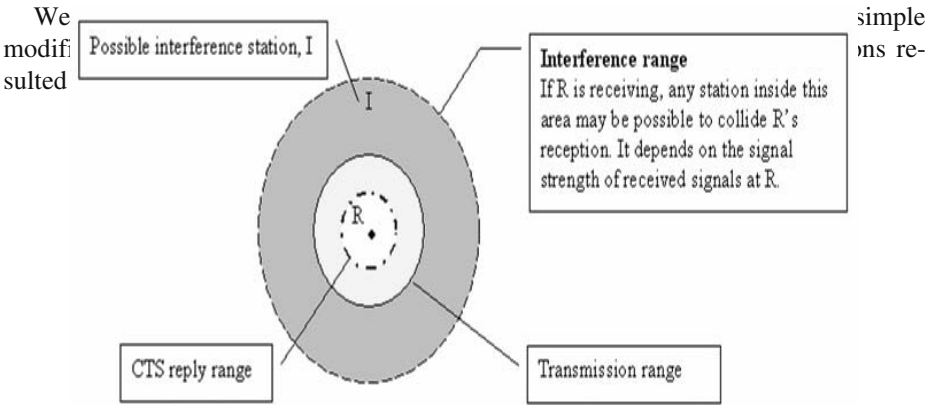


Fig. 4. The illustration of transmission range, interference range, and CTS reply range.

4 The AMAC Mechanism

The AMAC (Adaptive MAC) protocol modifies the IEEE 802.11 RTS/CTS handshake on transmitter (RTS) and receiver (CTS), respectively. The principle of our modification is "What you wish done to yourself, do to others; what you do not wish done to yourself, do not do to others." In reception phase, we make the hardware's best to receive but estimate for the probability of successful reception to prevent from

useless weak signal transmission (compared to hidden terminals' signal) which could be collided with hidden terminals' transmissions near the receiver. In transmission phase, we not only consider the most transmission opportunity but also give neighboring nodes chances to transmit or receive.

4.1 The Receiver Side Control Mechanism

Inspired from the CCR scheme, we propose a simple mechanism that dynamically adjusts the value of "CTS_REPLY_THRESHOLD" according to the historical neighboring medium usage status. If the receiver has some information about the interfering nodes, it can estimate the interference probability, and set the appropriate CTS_REPLY_THRESHOLD. Note that the CCR scheme assigns the CTS reply range to be a fixed value, for example $0.56 \cdot R_{tx}$, while our goal is aimed to dynamically adapt the CTS reply range according to the channel status at receiver side.

Mobile nodes periodically sense the medium status and record the sensed signal strength as shown in Figure 5. Let mobile nodes sense the medium status every *sense_signal_interval* micro seconds, total number of sensing is *sense_times*. The sensed signal strength will be recorded and the oldest recorded information will not be kept if the number of sensed records exceeds *sense_times*. Note that we only record the sensed signal strength transmitted inside the interference range but outside the receiving range, i.e. the signal transmitted inside the gray area of Figure 4. The recorded signal strength will be used to compute the CTS_REPLY_THRESHOLD, which concerns the potential transmissions that can cause interferences.

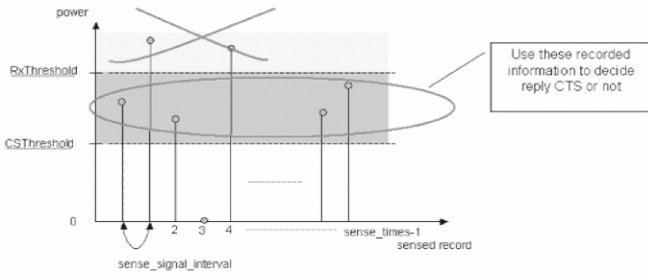


Fig. 5. Sense the medium periodically.

The considerations of CTS reply include two limitations. One is the estimation of whether interference will happen or not; another is the probability of interference if it happens. With the first limitation, we calculate the average of recorded signal strengths not equal to zero as equation (1), where Δ_i is used to identify whether P_i is needed to be included or not; n is the *sense_times*; P_i is the sensed signal strength between the threshold of receiving and of carrier sensing, and the *Capture_Threshold* is the capability the receiver can capture signal from noise. Usually the *Cap-*

ture_Threshold is 10. Hence, the CTS_REPLY_THRESHOLD could reflect average interference status.

$$\Delta_i = \begin{cases} 1, & \text{carrier_sense_threshold} \leq P_i < \text{receive_threshold} \\ 0, & \text{otherwise} \end{cases}$$

$$\text{CTS_REPLY_THRESHOLD} = \left(\frac{\sum_{i=0}^{n-1} P_i * \Delta_i}{\sum_{i=0}^{n-1} \Delta_i} \right) * \text{Capture_Threshold} \quad (1)$$

With the second limitation, we consider the probability of the reception be collided (p_{collided}). Again we use the past recorded signal strength. This probability is obtained as equation (2), where Δ_i is used to identify whether P_i larger than P_{RTS} or not; the P_{RTS} is the received signal strength of the RTS control frame.

$$\Delta_i = \begin{cases} 1, & P_i > P_{\text{RTS}} \\ 0, & \text{otherwise} \end{cases}$$

$$p_{\text{collided}} = \frac{\sum_{i=0}^{n-1} \Delta_i}{n} \quad (2)$$

After receiving RTS successfully, the intended receiver first checks if the power of the received RTS is larger than the computed CTS_REPLY_THRESHOLD; if larger, then he replies CTS back; if not larger, then he reply CTS with probability $1 - p_{\text{collided}}$.

This is because collisions happen at receivers. It is better for receivers to take the responsibility for monitoring the signal statuses and to estimate admitting receptions. So, if a node I, which is located outside the transmission range of receiver R and transmits frequently, any transmitter S may experience a collision at R, even if the RTS/CTS handshake is presented. If R decides to only reply S when the receiving signal strength of S is higher than the calculated CTS_REPLY_THRESHOLD, derived from R's observation, the interference will be reduced. On the other hand, if neighboring nodes transmit infrequently, then the CTS_REPLY_THRESHOLD should not be hold as the CCR specified; the threshold should be relaxed. This can be done by our periodical computation so as to dynamically reflect the current situation.

This mechanism has two benefits. One is that it can adapt to surrounding channel status. If the channel is always busy or the channel is too noisy around the receiver, then the receiver will decide not to reply the CTS for a demand RTS request. Another benefit is that although the receiver may not reply a CTS back to the RTS requestor, it is still a good decision because the "weak" data transmission (compared to the stronger interference signal strength at receiver) may not be received by the receiver, due to large interference range. At the same time, this weak data transmission could still collide other nodes' receptions near the transmitter, just like the transmitter's data collided by other nodes at receiver. So this mechanism should be beneficial to wireless ad hoc networks.

4.2 The Transmitter Side Control Mechanism

As Section 2 stated, the CSMA/CA, which is a random access control mechanism, is contention-based. All mobile nodes have to contend for the channel to transmit. As explained before, due to large interference range, a transmission may be collided by another transmission and collide other's transmission. There is a need to control unnecessary or excessive transmission, so we propose the transmitter side control mechanism that adapts to neighborhood situations. And also could improve some degree of fairness.

Before actually sending out a control frame, the RTS, the mobile node first checks the neighbors' activities, which is gotten from the periodically sensing the receiver side does it. Let δ_i denote whether neighboring nodes transmit at i^{th} carrier sensing or not; N is the total number of neighboring nodes' transmission activities; and *neighbor_tx_ratio* is defined as the portion the total number of times neighboring nodes transmit in *sense_times* periodically sense as equation (3) shows.

$$\delta_i = \begin{cases} 1, & P_i > \text{carrier_sense_threshold} \\ 0, & \text{otherwise} \end{cases}$$

$$N = \sum_{i=0}^{n-1} \delta_i \quad (3)$$

$$\text{neighbor_tx_ratio} = \frac{N}{n}$$

If *neighbor_tx_ratio* is less than a *neighbor_tx_threshold*, and the mobile node himself sent recently, this may imply that neighboring nodes hardly contend for the channel. So we force the mobile node to backoff to release the medium access for neighboring nodes.

Another case for this is that the neighboring nodes' silence is because of no data to send or being inhibited by RTS/CTS. So after backoff, mobile node observes whether the total number of neighboring nodes' transmission activities increases or not. If yes, the *neighbor_tx_threshold* is increased. Otherwise, it may imply that the neighboring nodes really have no data to send out, and the mobile node can speed up its transmission. In this case, the *neighbor_tx_threshold* is decreased. Here we use heuristic method to change the value of *neighbor_tx_threshold*.

By our both receiver side and transmitter side AMAC control, we can reduce fragile frame that the power strength of receiving signal at receiver is low or easily be collided with hidden terminals in the interference range, by not responding CTS frame, and won't collide with other's reception if the node transmits. We can also achieve MAC level fairness in some degrees by means of our transmitter side control to give other nodes chances to transmit or to receive.

5 Performance Evaluation

We use the NCTUns simulator [13] to justify our proposed modification of the IEEE 802.11 MAC protocol. The MAC protocol of the NCTUns is ported from NS-2 network simulator which implements the complete IEEE 802.11 standard MAC protocol DCF to accurately model the contention of nodes for the wireless medium. All nodes communicate with identical, half duplex, wireless radios that are modeled after the commercially available 802.11(b)-based Wave-Lan wireless radios, which have a bandwidth of 11 Mb/s and a nominal transmission range of 250 meters with the carrier sensing range is 550m. Note here that the value of CTS_REPLY_THRESHOLD is chosen as a receiving power at a receiver which is $0.56 \cdot R_{tx}$ away from the transmitter in the following simulations.

5.1 The Simulation of 1 TCP Flow via Multihop Transmission

We first setup a simple simulation to analyze the relationship between the interferences and the distance of transmitter-receiver. Each node is identical. The path loss model is set to Two-Ray Ground model; DSDV routing protocol is used; the traffic generator is a greedy TCP. The topology of this simulation is a 7 nodes chain topology as shown in Figure 6.

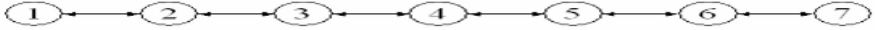


Fig. 6. The topology of 1 TCP flow simulation.

The distance between any two neighboring nodes is equal, let $d(i, i+1)$ be the distance between node i and node $i+1$. We adjust the distance of $d(i, i+1)$ from 100m to 200m to observe the throughput variation. As the distance between each node increases, the number of hops increases. The simulation result is shown in Figure 7.

We can see that when $d(i, i+1)$ is less than 125m, the throughput of IEEE 802.11 is high. This is because with such distances, the transmission from node 1 to node 7 only involves 3 hops and each sender, i.e. the node 1, node 3, and node 5 can sense each other, therefore can avoid most interferences. However, once the distance $d(i, i+1)$ is bigger than 125m (i.e. more than 3 hops from node 1 to 7), the throughput degrades significantly. This confirms that the IEEE 802.11 was primordially designed for WLAN, was not designed for multihop ad hoc network, the IEEE 802.11 MAC cannot function well in such networks.

We also take a look of the CCR scheme, the throughput was not good in this scenario, it is because the CCR is over conservative that cannot reflect the neighboring nodes' situations so that they only accept the received signal strength of RTS control frame greater than Conservative CTS_REPLY_THRESHOLD. With $d(i, i+1)$ larger than 141m ($250m \cdot 0.56$), the CCR won't reply CTS. Hence, no transmissions occur.

Now, let us consider our proposed scheme, AMAC. The throughput is not good in contrary to IEEE 802.11 MAC protocol when the $d(i, i+1)$ is 100m and 125m. This is

because our control in transmitter and receiver will reduce the transmission rights. However, when the distance is larger, the performance of AMAC is better than the other two MAC protocols. This is because our scheme can decrease the probability of collisions. As a metric define in [7], we define the "data frame collision ratio" as the portion of data frames transmitted at the MAC layer which are collided at the intended receiver due to collision; the retransmission data frame is viewed as a new data frame. We also only count the data frame collision ratio of unicast data frames, thus exclude routing packets that are broadcasting based. The simulation result of data frame collision ratio is presented in Figure 8.

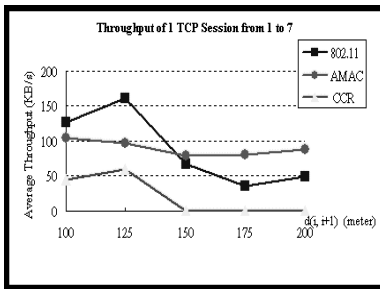


Fig. 7. The data frame collision ratio of 1 TCP session from node 1 to node 7.

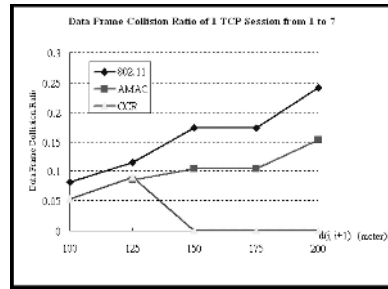


Fig. 8. The throughput of 1 TCP session from node 1 to node 7.

From Figure 8, we can see that the IEEE 802.11 suffers from more serious colliding than CCR and AMAC. It is also a hint that the contention manner of CSMA/CA may not suit to multihop connections. Every node constructive to compete the medium interferes other nodes. Eventually, every node interferes every node. Therefore, it is evidenced that the current version IEEE 802.11 not function well in multihop wireless ad hoc networks. There should have some efforts to overcome collision problems resulted from large interference range.

5.2 The TCP Instability Problem Simulation

In [8], they revealed the TCP instability problem in IEEE 802.11 multihop networks. This problem is resulted from the insufficient access control of IEEE 802.11 MAC. Here we redo the same simulation as [8] stated. The simulation topology is still a chain topology as Figure 6 shown with $d(i, i+1)$ is 200m. The parameters are also the same as section 5.1 used. The traffic generator is a TCP sender that always has data to send out. DSR routing protocol is used. The traffic is from node 1 to node 5. The simulation result is shown in Figure 9.

From the figure, we can see that our proposed scheme can eliminate the TCP instability problem. The reason is that the senders controlled by our AMAC adjust their transmission rate, which can fit well with this chain multihop transmission.

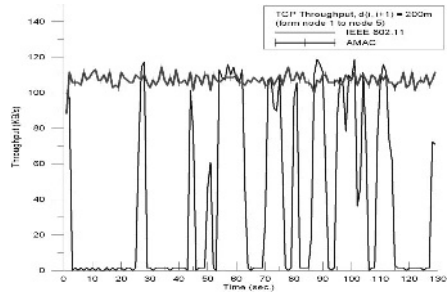


Fig. 9. The TCP throughput.

5.3 The Large Interference Range Simulation

We now estimate the influences of large interference range. The simulation topology is shown in Figure 10. We let the distance $d(i, i+1)$ 200m, except $d(2, 3)$ is well calculated to 355 meters so that $d(1, 3)$ is 555 meter that node 3 and node 1 cannot sense each other, but node 3 can interfere node 2. CCR is not applicable here.



Fig. 10. The topology of interference simulation.

The scenario is that node 1 sends CBR/UDP traffic to node 2 as "Flow 1", the CBR packet size is 1024 bytes and the packet rate is 500pps, node 3 sends VBR/UDP traffic to node 4 as "Flow 2", where the VBR packet size is also 1024 bytes (Poisson dist.). VBR packet rate is increasing by time to observe how it interferes with flow 1.

5.4 Random Topology Simulation

Finally, we simulate in a random topology scenario. 100 nodes are randomly placed in a 1000m * 1000m area. DSR routing protocol is used. We randomly select 4 source-destination pairs to send UDP/CBR traffic. The packet rate is 500 pps with packet size 1024 bytes.

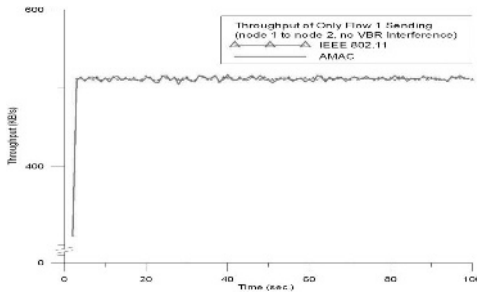


Fig. 11. The throughput of only node 1 sending.

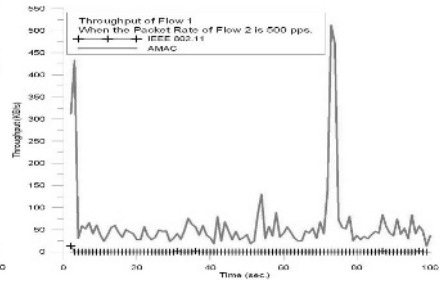


Fig. 12. The throughput comparison when the packet rate of node 3 is 500 pps.

The simulation results are shown in Figure 13 and Figure 14. We can see that the throughput of AMAC is higher than the IEEE 802.11 and CCR. This is because our scheme can reduce the collisions, thus improve the throughput. The CCR scheme can really eliminate the data corruption, but the throughput is not very good because it needs more hops to reach destination.

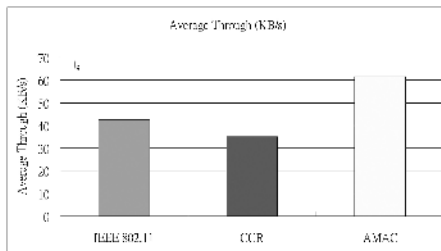


Fig. 13. throughput for the random topology.

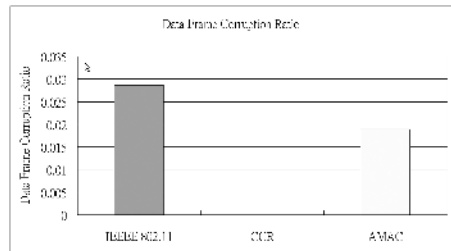


Fig. 14. Data frame corruption ratio.

6 Conclusion

In the multihop wireless ad hoc networks, the performance of IEEE 802.11 MAC degrades dramatically. The networks suffer from more serious hidden terminal problem than the WLAN because of large interference range. In this paper we are inspired from previous analysis of [7] to propose an adaptive MAC (AMAC) scheme by modifying the IEEE 802.11 MAC RTS/CTS handshake. We add two control mechanisms on transmitter and receiver with the objective of reducing probability of collisions and reducing the number of collisions. The simulation results show that our modification used in the multihop wireless ad hoc networks outperforms the IEEE 802.11 MAC. Besides, there are some issues for future work, for example the estimation model at

receiver may work well and the optimum value of parameters used in both transmitter side and receiver side.

References

1. Antoine Mercier, Pascale Minet, Laurent George, and Gilles Mercier, "Adequacy between multimedia application requirements and wireless protocols features," *IEEE Wireless Communications*, vol.9 No.6, pp. 26–34, December 2002.
2. Ramiro Jordan and Chaouki T. Abdallah, "Wireless communications and networking: an overview," *IEEE Antenna's and Propagation Magazine*, vol. 44 no. 1, February 2002.
3. IEEE, 1999, Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications, *IEEE Std. 802.11*.
4. Crow, B.P.; Widjaja, I.; Kim, L.G.; Sakai, P.T., "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine*, Volume: 35 Issue: 9, pp.116–126, Sept. 1997.
5. L. Kleinrock and F. Tobagi, "Packet switching in radio channels, part II-the hidden terminal problem in carrier sense multiple access and the busy tone solution," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1417–1433, Dec. 1975.
6. P. Karn, "MACA-A New Channel Access Method for Packet Radio," in *Proc. 9th ARRL Computer networking Conference*, 1990.
7. Kaixin Xu, Mario Gerla, and Sang Bae, "How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks?" *GLOBECOM 2002 – IEEE Global Telecommunications Conference*, no. 1, pp. 72–77, November 2002.
8. Xu, S.Saadawi, T. "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?" *IEEE Communications Magazine*, Volume: 39 Issue: 6, pp. 130–137, June 2001.
9. Jinyang. Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris, "Capacity of ad-hoc wireless networks," *Proceedings of ACM MOBICOM 01*, pp. 61–69, July 2000.
10. Joao L. Sobrinho, A. S. Krishnakumar, "Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks," *IEEE Journal on Selected Areas in Communications*, no. 8, pp. 1353–1368, August 1999.
11. VINT Group, UCB/LBNL/VINT network simulator-ns (version 2), <http://www.isi.edu/nsnam/ns>.
12. The CMU Monarch Project, Wireless and mobility extension to ns, <http://www.monarch.cs.cmu.edu>.
13. S.Y. Wang, C.L. Chou, C.H. Huang, C.C. Hwang, Z.M. Yang, C.C. Chiou, and C.C. Lin, "The Design and Implementation of the NCTUns 1.0 Network Simulator", *Computer Networks*, Vol. 42, Issue 2, pp. 175–197, June 2003.

A Distributed Algorithm for Bandwidth Allocation in Stable Ad Hoc Networks

Claude Chaudet¹, Isabelle Guérin Lassous¹, and Janez Žerovnik^{2,3*}

¹ Inria Ares Team, Laboratoire Citi, Insa de Lyon, 21, avenue Jean Capelle, 69621 Villeurbanne Cedex, France

Claude.Chaudet@insa-lyon.fr, Isabelle.Guerin-Lassous@inrialpes.fr

² University of Maribor, FS, Smetanova 17, 2000 Maribor, Slovenia

³ Institute of Mathematics, Physics and Mechanics, Department of Theoretical Computer Science, Jadranska 19, 1111 Ljubljana, Slovenia.
janez.zerovnik@uni-lj.si

Abstract. We propose a distributed algorithm for allocating bandwidth in stable ad hoc networks. After having discussed the problem of bandwidth allocation in such networks, we define a sequence of feasible solutions to this problem. This sequence has the property to be an increasing sequence in terms of overall used bandwidth. After a theoretical analysis of the sequence, we design a distributed algorithm based on this sequence. We test our algorithm by simulations on different topologies. Results obtained on geometric random graphs are presented here. We compare our solutions with the optimal solution in terms of global bandwidth allocation that presents the smallest standard deviation and with the the fairest solution regarding to max-min fairness. The simulations show that the global used bandwidth is less than 25% from optimality in the worst case and the standard deviation is the smallest of the three.

Introduction

Today, most of the existing wireless radio networks (GSM, WiFi) are infrastructure based networks. A fixed base station manages the transmissions in a certain geographic zone corresponding to its transmission range. Mobiles in this area need to be connected directly to this base station in order to communicate.

Mobile ad hoc networks are an evolution of these wireless networks in which no fixed infrastructure is needed. Mobiles communicate directly between each other, without the need for a base station. To enable these communications, any mobile should be able to perform routing for the others. The MANET (Mobile Ad hoc NETWORKS) working group has been created at the IETF in order to standardize a routing protocol for ad hoc networks. This working group has arrived in the final phase of its researches and is about to decide what the best effort routing protocol for these networks will be.

* Supported in part by the Ministry of Education, Science and Sport of Slovenia.

In these networks, the radio medium is shared by all the mobiles. Many techniques are available to manage multiple access to the medium, from centralized or semi-centralized ones (time slot, frequency or spreading code allocation for example) to totally distributed ones (CSMA type for instance). Due to the infrastructure-less nature of ad hoc networks, the distributed medium access protocols seem more suited.

As most of the available wireless interface cards implement the IEEE 802.11 [1] standard, most of the actual ad hoc networks rely on it. This standard provides a totally distributed mode for the medium access part based on a CSMA / CA mechanism: the mobiles communicate on the same frequency and as long as one mobile emits, it prevents all of its neighbors from transmitting. Otherwise if two nearby mobiles emitted simultaneously on the same frequency, the two signals would interfere and there would be a high probability that none of them results in a successful transmission. As long as some bandwidth remains available, the MAC protocol manages to share quite efficiently the medium. As long as there is no congestion, mobiles can freely access the medium (in accordance with the 802.11 MAC protocol) and use whatever share of the resources they need. When the medium capacity is exceeded, the protocol behavior and the subsequent medium share become unpredictable.

Most of the proposed protocols for bandwidth management in ad hoc networks have an *a posteriori* approach, i.e. the bandwidth is managed only when congestion points appear. Protocols like INSIGNIA [2], SWAN [3] or HMP [4] do use such an approach. They provide notification, degradation or/and re-routing in order to react to the appearance of congestion. These approaches usually need additional communication between mobiles when the network already suffers from congestion. Therefore, returning to a stable situation can take some time. An alternative would be to manage the bandwidth *a priori*. If each mobile controls the use of its bandwidth rather than trying to use the whole radio medium, it will be able to prevent most of the congestion situations from appearing. It is difficult in such networks to avoid any congestion due to mobility: the control applied by one mobile in one configuration may not be adapted anymore in another configuration obtained by its mobility and may lead to the appearance of congestion. But an *a priori* solution may prevent from most of the congestion and may give efficient results in terms of bandwidth management as long as the network is quite stable with a low mobility.

In these networks, mechanisms like bandwidth reservation are essentially provided for constrained traffic, for example real-time flows. If the admission control is well suited to the characteristics of the network, reservations can prevent congestion from appearing. Best effort traffic is usually not limited by any mechanism and can thus easily overlap on the privileged traffics' bandwidth share, making the guarantees fragile. One possible solution to this problem is to allocate a constant amount of bandwidth for best effort traffic but this does neither take into account the resources required for the privileged traffic nor the topology of the network. Such solution often leads to a sub-optimal use of the network resources. An alternative is to allocate bandwidth to best effort traffic

according to the properties of the network, i.e. the topology and the bandwidth available to each mobile. This assignment is done so that no saturation appears on any mobile. Finding such a solution while maximizing at the same time the overall used bandwidth in the network is equivalent to a fractional packing problem. Algorithms solving this problem are essentially sequential and difficult to adapt to a distributed setting. Moreover, such solutions maximize the total used bandwidth without guaranteeing any fairness among the mobiles. Not providing a minimum amount of bandwidth for each mobile in the network may lead to serious imbalance and to a bad use of the network.

In this article, we propose a distributed algorithm to allocate bandwidth to each mobile according to the topology of the network and the available bandwidth on each mobile for stable ad hoc networks. The algorithm guarantees a non null minimum bandwidth to each mobile. With this algorithm, each mobile computes the bandwidth it can use in order to avoid saturating its capacity or its neighbors'. With such an algorithm, congestion is less likely to appear in the network.

In Sect. 1, we give the used model for ad hoc networks and the bandwidth allocation problem. A simple bandwidth allocation is given in Sect. 2. From this allocation, we design a sequence of feasible allocation in Sect. 3. Each term of the sequence is a new bandwidth allocation that is more efficient in terms of global used bandwidth than the previous terms in the sequence. Different properties of the sequence are described in this section. From this sequence, we design a distributed algorithm that allocates bandwidth to each mobile of the network in Sect. 4. At each step of the algorithm, each mobile needs only to know the minimum remaining bandwidth and the maximum degree in its neighborhood. In Sect. 4.3, we present the results obtained by simulation on geometric random graphs. This work is on going and we evaluate in this article the quality of our allocation in terms of overall used bandwidth and of fairness among the mobiles. The obtained results should allow us to decide if our algorithm can be converted in an efficient bandwidth allocation protocol for ad hoc networks. In the conclusion, we discuss the points to solve for the protocol version and how to integrate mobility in our solution.

1 The Model and the Problem

We model our ad hoc network by a vertex weighted graph $G(V, E, b)$ where:

- $V = V(G)$ is the set of vertices of the graph. One vertex in the graph represents one mobile in the network;
- $E = E(G)$ is the set of edges of the graph. There is an edge between two vertices whenever the two corresponding mobiles are able to communicate, i.e. are in each other's transmission range;
- b is a function which assigns positive real numbers to vertices, representing the capacity of the medium around each mobile.

A congestion appears whenever the capacity of the medium is exceeded in a certain region of the network. If we suppose that mobiles only share the medium

with their direct neighbors, ensuring that for each mobile in the network, the sum of its used bandwidth and of the bandwidth used by its neighbors does not exceed this mobile's capacity will ensure that there is no congestion at all in the network. Formally, if, for each mobile v in the network, we note $N[v] = \{v\} \cup \{u \mid uv \in E\}$ its closed neighborhood and $x(v)$ the amount of bandwidth that v can use, the problem can be expressed as:

$$\forall v \in V, \sum_{u \in N[v]} x(u) \leq b(v). \quad (1)$$

Maximizing at the same time the overall use of the network is equivalent to the following problem:

$$\max \sum_{v \in V} x(v) \quad s.t. \quad \forall v \in V, \sum_{u \in N[v]} x(u) \leq b(v).$$

This linear problem is known as a fractional packing problem. The problem can be solved by usual linear programming algorithms, and there are also faster approximation algorithms known (see [5] for details and for further references). These algorithms are sequential. A distributed algorithm for linear programming that obtains a $(1 + \varepsilon)$ -approximation in polylogarithmic number of communication rounds is given in [6]. Solutions to the fractional packing problem maximize the total bandwidth used in the network and give no guarantee on the minimum bandwidth for each mobile (i.e. the bandwidth may be null on some mobiles). This may have an impact on the good running of the network.

We propose a distributed algorithm that computes a set of values $\{x(v)\}_{v \in V}$ solution to the constraints (1). The algorithm is based on the topology of the network and the available bandwidth of each mobile. Before presenting the algorithm, we define the notation and formally evaluate the features of the solution in Sect. 2 and Sect. 3.

2 The Basic Lemma

We will use the following notations:

- $N[v]$ is the closed neighborhood of the vertex v ,
- $N(v)$ is the open neighborhood of the vertex v ($N(v) = N[v] \setminus \{v\}$),
- $d(v)$ is the degree of vertex v ,
- $\mathbf{x} = (x(v), v \in V)$ is the vector of values we are trying to compute, i.e. the solution of (1), i.e. from a networking point of view, the allocated bandwidth to the mobiles,
- $\mathbf{b} = (b(v), v \in V(G))$ is the vector of the "capacities" of the vertices, i.e. from a networking point of view, the available bandwidth for each mobile,
- $\Delta_1(v) = \max_{u \in N[v]} d(u)$, i.e. the maximum degree over the closed neighborhood of v (v included),

- $b_1(v) = \min_{u \in N[v]} b(u)$, i.e. from a networking point of view, the minimum bandwidth over the closed neighborhood of v (v included).

Lemma 1. *If for every node v in the graph, $x(v) = \frac{b_1(v)}{\Delta_1(v)+1}$, then $\mathbf{x} = (x(v), v \in V)$ is suitable to the constraints (1).*

Proof. Recall from definitions of Δ_1 and b_1 that $u \in N[v]$ implies that $d(v) \leq \Delta_1(u)$ and $b(v) \geq b_1(u)$. Therefore,

$$\begin{aligned} \sum_{u \in N[v]} x(u) &\leq \sum_{u \in N[v]} \frac{b_1(u)}{\Delta_1(u) + 1} \leq \sum_{u \in N[v]} \frac{b(v)}{d(v) + 1} \leq \frac{d(v) + 1}{d(v) + 1} b(v) \\ &\leq b(v). \end{aligned}$$

It means that \mathbf{x} can be used as an initial solution to the constraints (1) and is a feasible bandwidth allocation.

3 A Sequence of Feasible Vectors

In Sect. 2, we have found a vector of values that respects the constraints defined by (1). In this section, we are going to show that if we iterate the process using the remaining bandwidth at each mobile after having computed \mathbf{x} , then we still have a solution to the problem and we increase the overall used bandwidth in the network.

3.1 Sequence Definition

We will consider the following sequences of vectors:

- $x^{(i)}(v)$ represents the allocated resources amount for the node v at the i^{th} step of the sequence,
- $e^{(i)}(v)$ represents the remaining capacity of the node v at the i^{th} step considering what its neighbors have taken at this step.

These values are initialized as follows:

$$\begin{aligned} x^{(0)}(v) &= \frac{b_1(v)}{\Delta_1(v) + 1}. \\ e^{(0)}(v) &= b(v) - \sum_{u \in N[v]} x^{(0)}(u). \end{aligned}$$

Then the passage from step i to step $i + 1$ is done on the following way:

$$\begin{aligned} x^{(i+1)}(v) &= x^{(i)}(v) + \frac{1}{\Delta_1(v) + 1} \cdot \min_{u \in N[v]} e^{(i)}(u). \\ e^{(i+1)}(v) &= b(v) - \sum_{u \in N[v]} x^{(i+1)}(u). \end{aligned}$$

3.2 Sequence Properties

Lemma 2. *All the terms of this sequence respect the constraints defined by (1), i.e.*

$$\forall v \in V, \forall i \in \mathbb{N}, \sum_{u \in N[v]} x^{(i)}(u) \leq b(v).$$

Proof. We can write, $\forall v \in V, \forall i \in \mathbb{N}$,

$$\forall u \in N[v], \min_{w \in N[u]} e^{(i)}(w) \leq e^{(i)}(v).$$

$$\Rightarrow \sum_{u \in N[v]} \left(\min_{w \in N[u]} e^{(i)}(w) \right) \leq |N[v]| \times e^{(i)}(v).$$

As $|N[v]| > 0$, we can write:

$$\frac{\sum_{u \in N[v]} \left(\min_{w \in N[u]} e^{(i)}(w) \right)}{|N[v]|} \leq e^{(i)}(v).$$

As $\forall u \in V, 1 + \Delta_1(u) \geq 1 + d(v)$ for any $v \in N[u]$ and $1 + d(v) = |N[v]|$, then

$$\sum_{u \in N[v]} \min_{w \in N[u]} e^{(i)}(w) \times \frac{1}{\Delta_1(u) + 1} \leq e^{(i)}(v).$$

Therefore

$$\sum_{u \in N[v]} (x^{(i+1)}(u) - x^{(i)}(u)) \leq b(v) - \sum_{u \in N[v]} x^{(i)}(u).$$

$$\Rightarrow \sum_{u \in N[v]} x^{(i+1)}(u) \leq b(v).$$

We have proved that $\forall v \in V, \forall i \in \mathbb{N}^*, \sum_{u \in N[v]} x^{(i)}(u) \leq b(v)$.

Thanks to Lemma 1, we know that

$$\forall v \in V, \sum_{u \in N[v]} x^{(0)}(u) \leq b(v).$$

Therefore,

$$\forall v \in V, \forall i \in \mathbb{N}, \sum_{u \in N[v]} x^{(i)}(u) \leq b(v).$$

It means that each element of the sequence is a solution to the constraints (1) and is a feasible bandwidth allocation.

Lemma 3. *If the vector \mathbf{b} contains no zero then every allocated value is non-zero, i.e. $\forall v \in V, \forall i \in \mathbb{N}, x^{(i)}(v) > 0$.*

Proof. Straightforward with the definition of $x^{(0)}(v)$ and Lemma 2.

It means that each term of the sequence corresponds to a bandwidth allocation where no mobile has a bandwidth equal to 0.

Lemma 4. *The sequence $(x^{(i)}(v))_{i \in \mathbb{N}}$ is convergent.*

Proof. We can write, using the definition of $e^{(i)}(v)$ and Lemma 2, that $\forall v \in V, \forall i \in \mathbb{N}, e^{(i)}(v) \geq 0$. Therefore, the sequence $(x^{(i)}(v))_{i \in \mathbb{N}}$ is monotone increasing. As a monotone increasing and bounded sequence always converges, and as $(x^{(i)}(v))_{i \in \mathbb{N}}$ is bounded by $b(v)$, $(x^{(i)}(v))_{i \in \mathbb{N}}$ converges.

It means that each term of the sequence has an overall bandwidth greater than the previous terms in the sequence and that the sequence tends to a solution that has the maximum global bandwidth within this sequence.

Lemma 5. *A node v will reach a null remaining bandwidth at step i ($e^{(i)}(v) = 0$) if and only if this node has the minimum remaining bandwidth and the maximum degree among its neighbors at step $i - 1$.*

Proof. Assume that a node v has a non null remaining bandwidth at step $i - 1$ (i.e. $e^{(i-1)}(v) > 0$) and does not have the maximum degree among its neighbors (i.e. $d(v) < \Delta_1(v)$ and $\exists z \in N(v)$ s.t. $d(z) = \Delta_1(v)$), by definition :

$$\begin{aligned} e^{(i)}(v) &= e^{(i-1)}(v) - \sum_{u \in N[v]} \frac{\min_{w \in N[u]} e^{(i-1)}(w)}{\Delta_1(u) + 1} \\ &= e^{(i-1)}(v) - \left(\sum_{u \in N[v] \setminus \{z\}} \frac{\min_{w \in N[u]} e^{(i-1)}(w)}{\Delta_1(u) + 1} \right) - \frac{\min_{w \in N[z]} e^{(i-1)}(w)}{\Delta_1(z) + 1}. \end{aligned}$$

As $\forall u \in N[v], d(v) \leq \Delta_1(u)$ and as $\forall w \in N[u], \min_{w \in N[u]} e^{(i-1)}(w) \leq e^{(i-1)}(v)$, then we have:

$$\frac{\min_{w \in N[u]} e^{(i-1)}(w)}{\Delta_1(u) + 1} \leq \frac{e^{(i-1)}(v)}{d(v) + 1}.$$

And as $\Delta_1(z) \geq d(z) > d(v)$:

$$\frac{\min_{w \in N[z]} e^{(i-1)}(w)}{\Delta_1(z) + 1} < \frac{e^{(i-1)}(v)}{d(v) + 1}.$$

Injecting these two expressions in the previous one, we obtain:

$$e^{(i)}(v) > e^{(i-1)}(v) - d(v) \cdot \frac{e^{(i-1)}(v)}{d(v) + 1} - \frac{e^{(i-1)}(v)}{d(v) + 1} > 0.$$

Therefore, if a node has a non null bandwidth and does not have the maximum degree among its neighbors at a certain step, it will not reach its allocation limit at the next step. It is easy to show the same property for any node v that has a non null remaining bandwidth at a given step and has a minimum remaining bandwidth greater than one of its neighbors'. Therefore a node can reach its limit at a given step only if it had the minimum remaining bandwidth and the maximum degree among its neighbors at the previous step.

Proving that a node with minimum remaining bandwidth and maximum degree among its neighbors at some step reaches its limit at the next step is straightforward from the definition of $x^{(i)}$.

This lemma characterizes the nodes that can reach a null remaining bandwidth. When they reach this state, their allocated bandwidth won't increase anymore in the next steps of the sequence and will remain constant. Note that the neighbors of such nodes will also reach the limit of their allocated bandwidth: they can't take more bandwidth in the next steps of the sequence otherwise they will saturate the capacity of the neighbor nodes that have a null remaining bandwidth.

3.3 Quality of the Solution

Lemma 6. *Every node has at least a neighbor whose free bandwidth converges towards 0:*

$$\forall v \in V, \exists u \in N[v] \text{ s.t. } \lim_{i \rightarrow +\infty} e^{(i)}(u) = 0.$$

Proof. As $(x^{(i)}(v))_{i \in \mathbb{N}}$ converges, using the definition of the terms of the sequence: $x^{(i+1)}(v) = x^{(i)}(v) + \frac{1}{\Delta_1(v)+1} \min_{u \in N[v]} e^{(i)}(u)$, we can see that the number $x^{(i+1)}(v) - x^{(i)}(v) = \frac{1}{\Delta_1(v)+1} \min_{u \in N[v]} e^{(i)}(u)$ converges towards 0. Therefore, $\min_{u \in N[v]} e^{(i)}(u)$ converges towards 0.

Lemma 7. *The sequence $(x^{(i+1)}(v) - x^{(i)}(v))_{i \in \mathbb{N}}$ is decreasing.*

Proof. As the sequence $(x^{(i)}(v))_{i \in \mathbb{N}}$ is increasing, then the sequence $(e^{(i)}(v))_{i \in \mathbb{N}}$ is decreasing. Then:

$$\forall u \in N[v], e^{(i)}(u) \leq e^{(i-1)}(u).$$

Therefore,

$$\exists z \in N[v] \text{ s.t. } \min_{u \in N[v]} e^{(i-1)}(u) = e^{(i-1)}(z) \geq e^{(i)}(z) \geq \min_{u \in N[v]} e^{(i)}(u).$$

And

$$(x^{(i+1)}(v) - x^{(i)}(v)) \leq (x^{(i)}(v) - x^{(i-1)}(v)).$$

From now on, we will note $X(v)$ the limit of the sequence $(x^{(i)}(v))_{i \in \mathbb{N}}$, X the limit vector solution and $E(v)$ the remaining bandwidth at node v with the solution X .

Lemma 8. *X is Pareto-efficient.*

Proof. We consider the order \leq where \leq is the natural order on \mathbb{R}^N , i.e. $x \leq y$ iff $x_i \leq y_i, \forall i \in [1, N]$. A solution is Pareto-efficient [7] if it is maximum in the sense of \leq .

Assume we have a vector S that respects the constraints (1) and so that $S > X$ according to the order previously defined. This means that there exists a node v so that $S(v) > X(v)$. According Lemma 6, there exists a node $u \in N[v]$ that has reached its bandwidth allocation limit. This means that $\sum_{w \in N[u]} X(w) = b(u)$. As $S(v) > X(v)$ and as S respects the constraints (1), necessarily there exists a node $z \in N[u] \setminus \{v\}$ such that $S(z) < X(z)$. Therefore we can not have $S > X$.

3.4 Convergence Speed

Not much can be said on this algorithm convergence speed. The speed is highly dependent on the graph topology and on the bandwidths available at each node.

On one hand, for a regular graph with uniform weights ($\forall v \in V, b(v) = b$), the algorithm finds the optimal solution for the fractional packing problem at the first step (i.e. $\forall v \in V, x^{(0)}(v) = \frac{1}{1+\Delta}b$ and $e^{(0)}(v) = 0$).

On the other hand, we can find networks configurations that result in a sequence that converges with an infinite number of steps. For example, consider a chain of three nodes. Nodes of index 1 and 3 can communicate with node of index 2 but not between themselves. Assume that $b(1) = b(3) = 2$ and $b(2) = 3$. After one step, the remaining bandwidth at each node is exactly divided by 3 ($2/3$ for nodes 1 and 3 and 1 for node 2) and the bandwidth keep the same ratios between each other. Therefore, all further steps will lead to a configuration in which the ratio between the bandwidths is the same as initially and hence the number of steps to converge is, in this configuration, infinite.

Now, if we consider the same configuration with $b(1) = b(3) = 2 \cdot b(2)$, the algorithm converges in a single step.

As we will show in Sect. 4.3, simulations give good results in the number steps of our algorithm needs to converge to a value near the limit of the sequence, whatever the network may be.

4 A Distributed Algorithm for Bandwidth Allocation

4.1 Base Algorithm

The following distributed algorithm Algorithm 1 is based on the sequence presented in Sect. 3. It computes an increasing sequence and each element of the sequence is a solution to the constraints (1). The algorithm locally computes in each vertex the sequence $(x^{(i)}(v))_{i \in \mathbb{N}}$ stored in the variable X . At the end

of the algorithm, X gives the bandwidth allocated at node v . The remaining bandwidth $(e^{(i)}(v))_{i \in \mathbb{N}}$ is stored in each vertex in the variable E . The information that needs to be gathered for the computation consists, at each step, in the values $(x^{(i)}(u), e^{(i)}(u))_{i \in \mathbb{N}}$ for each neighbor and the degree of each neighbor. These informations need to be broadcasted in two steps as the calculation of one sequence requires the updated value of the other sequence.

Algorithm 1 : Bandwidth allocation (at node v)

Input: the list of neighbors of v and $b(v)$ the available bandwidth at node v

Output: X the bandwidth allocated at node v

$E := b(v);$

$X := 0;$

while X is not constant **do**

send E and d to all neighbors;

receive $E(u)$ and $d(u)$ from all neighbors u ;

$X := X + \frac{1}{\Delta_1(v)+1} \min E(u)$; (minimum over closed neighborhood)

send X to all neighbors;

receive $X(u)$ from all neighbors;

$E := E - \sum X(u);$

4.2 Remarks on This Algorithm

- By Lemma 4, the algorithm clearly converges to a feasible solution X . Each term of the sequence $X^{(i)}$, represents a feasible solution for constraints (1) and gives the bandwidth that can be used by each mobile in the network. Each term $X^{(i)}$ gives an overall bandwidth greater than with the previous terms, but note that X is not always optimal in terms of global bandwidth as some examples presented in the next section will show.
- By Lemma 2, we know that the bandwidth allocated to the mobile with Algorithm 1 does not exceed the capacity of the network. This allocation is fair in the sense that no mobile overlaps on its neighbors resources amount. Moreover by Lemma 3 all mobiles do get some resources.
- By Lemma 6, we know that there is no "space left" in the resources once the algorithm is finished. That does not mean that all mobiles may use the total capacity of their wireless cards, but that means that each mobile has a neighbor in its closed neighborhood that has no space left. Thus, if a mobile wants to use more bandwidth than the one allocated then a congestion point will appear in its neighborhood because it will exceed the capacity of its neighbor that has no capacity remaining.
- By Lemma 7, we know that the difference between two consecutive values of X is decreasing. According to the quality of the solution we want to obtain, we can consider, at each step of the algorithm, that this difference becomes small enough to accept this solution and to not go further with the algorithm.

Therefore, instead of achieving X constant, we can stop the algorithm when the difference between two consecutive values of X is smaller than a given threshold. We discuss the impact of this threshold in the next section.

- We also designed and tested a simple optimized version. In this second version, the nodes also transmit the minimum bandwidth in their neighborhood. If this value is 0, then neighbors will know this node will not be able to take anymore bandwidth. Then, they don't have to take it into account in the max degree calculation in the next step. They nevertheless still need to be considered when looking at the minimum bandwidth in the neighborhood. Depending on the network topology, this optimized version may have a strong impact on the convergence with some configurations. We will compare the convergence speed of these two versions in Sect. 4.3.

4.3 Implementation Results of This Algorithm

Algorithm 1 has been implemented in C++. It has been tested on different kinds of configurations, like complete graphs, chains, rings, meshes and geometric random graphs. Due to space limitations, we will only present here the results obtained on geometric random graphs, as these graphs are in our opinion the most accurate representation of ad hoc networks. Other results can be found in the extended version of this article [8]. 50 runs of each test have been carried out on different sizes of the configurations and we give the average results.

The geometric random graphs we used to test our algorithm are generated by considering 100 nodes put in a 1000×1000 square and by increasing the communication range of each node. The communication area ranges from 130 to 200 meters. When the communication range increases, the average degree of the nodes also increases. Each node is given a random capacity (available bandwidth b) between 50 and 150.

To determine the quality of our algorithm, we compare our solution with a solution to the fractional packing problem and with the most fair solution respecting the constraints (1) regarding a max-min fairness criteria.

A solution to the fractional packing problem may be computed by classical linear programming using the simplex algorithm. This results in a solution respecting the constraints (1) that maximizes the overall used bandwidth but that is not necessarily fair, as it may assign 0 bandwidth to some nodes. As the optimal solution to the linear programming maximization problem defined by (1) is not unique, we needed to find the "most fair" of these solutions. We choose to use standard deviation between the allocated bandwidths as the fairness criteria. To find the most fair optimal solution according to this criteria, we need to minimize the standard deviation between the allocated bandwidths subject to the constraints defined by (1) to which we add another constraint: the sum of all the bandwidths has to be equal to the objective function value computed by linear programming. This quadratic problem was solved using OOQP [9] and the results are presented below. As the time complexity of quadratic programming relatively large and the computation is sequential, the properties of the solutions

can only be used as benchmarks for comparison and evaluation of the results of our algorithms.

We have also computed the most fair solution regarding to the max-min fairness criteria. This solution is often considered as the fairest allocation, as noted in [10,11]. As a quick overview, in the max-min fairest solution to a problem, no bandwidth can be increased without decreasing another bandwidth that is already lower than the first one. Our solution is not the max-min fairest solution (it is easy to find a counterexample). We compare our solution with the max-min optimal solution.

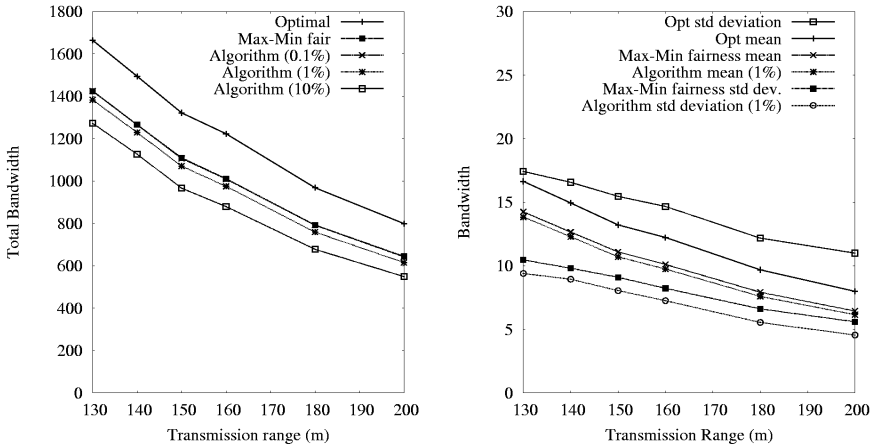


Fig. 1. Global bandwidth and statistics on the obtained solution on geometric graphs

The quality of the solution in terms of global bandwidth use is represented by the left plot of Fig. 1. First, we can notice that the global bandwidth decreases as the communication range increases. This fact can be explained by the increase of the constraints of the system due to the increase of the average degree of the nodes and by the constant number of nodes. The obtained bandwidth with our algorithm is between 18% and 25% smaller than the optimal one. The overall bandwidth with the max-min solution is slightly greater than with our solution. The increase in global bandwidth is around 9% for 200 meters when the threshold switches from 1% to 0.1%. On the other hand, this increase is much smaller with sparse networks (it is around 5% with 130 meters).

To give an indication on the fairness of the solution, we can compare the mean value and the standard deviation of the obtained solution with the two other solutions. These statistics, shown on the right plot of Fig. 1, show that our algorithm is quite fair with this configuration. The values allocated to the nodes range from around 4 to 24 in average for a communication range of 130 meters and from around 1 to 11 in average for a communication range of 200 meters. Our algorithm is fairer on dense networks than on sparse ones. As for

the other configurations, the standard deviation is around more than two times greater with optimal solution than with our solution. The difference between the max-min standard deviation and ours is more visible with these networks and is about 10%.

Lastly, we compare the convergence times of the initial algorithm and the modified algorithm as described in Sect. 4.2. Each result has been obtained with different stopping thresholds (as discussed in Sect. 4.2). Three values of this threshold have been considered: 10%, 1% and 0.1% (it means that for a threshold of 1% for instance, the algorithm stops as soon as two consecutive values of X differ for less than 1%).

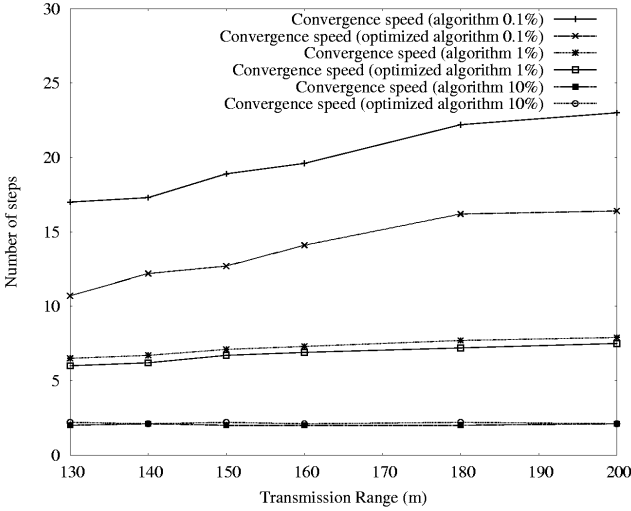


Fig. 2. Number of steps of Algorithm 1 on geometric graphs

Figure 2 shows that the convergence speed depends on the degree of the nodes, especially when the threshold is 0.1%. The maximum convergence speed is obtained with the initial algorithm with a threshold of 0.1% and is around 22 steps for the worst situation with a communication range of 200 meters. In this configuration, the convergence is reduced to 16 steps with the modified version. As for the other configurations, the speeding up increases with the refining of the threshold.

The results (accessible in [8]) obtained for chains, rings and meshes and complete graphs show that our algorithm is not far from the optimal in regular configurations (about 10% below the optimal global bandwidth). Moreover, the standard deviation is generally rather smaller than the standard deviation of the most fair optimal solution as well as the one of the max-min most fair solution.

By comparing these results, we can say that the number of nodes in the networks has a very limited impact on the quality of our solution (mean value, stan-

dard deviation and convergence speed) as long as the degrees remain constant. On the other hand, the degree is an important parameter and our algorithms perform better on dense graphs regarding fairness than on sparse ones and better on sparse graphs than on dense graphs regarding total bandwidth allocation and convergence speed.

Conclusion and Future Work

The results achieved by this algorithm are promising and encourage us to convert it into a protocol in order to test its stability and to determine its influence on congestion points appearance.

Nevertheless, this conversion will not be straightforward. Mobility is also a key issue. Ad hoc networks are supposed to be mobile networks and the protocol will have to take into account topology changes. One solution is for each mobile to check the feasibility of the allocation considering appearing links regarding the constraints (1). If one link violates one of the constraints, the involved mobiles on the link may then share the minimum initial allocated bandwidth between these two nodes. If this solution quickly backtracks to a feasible solution, it may lead to an allocation less fair than the proposed solution of this article. Another alternative is that a mobile suffering an over-allocation implies its neighbors in the return to a stable solution. This kind of mechanism would probably lead to a fairer allocation but it could take some time before stabilizing. Our future work is to investigate such solutions.

References

1. IEEE Standard for Information Technology Telecommunications and Information Exchange between Systems: Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (1997)
2. Lee, S.B., Ahn, G.S., Zhang, X., Campbell, A.T.: Insignia: An ip-based quality of service framework for mobile ad hoc networks. *Journal on Parallel and Distributed Computing* **60** (2000)
3. Ahn, G.S., Campbell, A.T., Veres, A., Sun, L.H.: SWAN: Service differentiation in stateless wireless ad hoc networks. In: *IEEE INFOCOM' 2002*, New York, USA (2002)
4. Lee, S.B., Campbell, A.T.: HMP: Hotspot mitigation Protocol for Mobile Ad Hoc Networks. In: *11th IEEE/IFIP International Workshop on Quality of Service*, Monterey, Canada (2003)
5. Plotkin, S.A., Schmoys, D.B., Tardos, E.: Fast approximation algorithms for fractional packing and covering problems. *Mathematics Of Operations Research* **20** (1995) 257–301
6. Bartal, Y., Byers, J.W., Raz, D.: Global optimization using local information with applications to flow control. In: *38th IEEE Symp. on Foundations of Computer Science*. (1997) 303–312
7. Feldman, A.M.: *Welfare Economics and Social Choice Theory*. Kluwer, Boston (1980)

8. Chaudet, C., Guérin Lassous, I., Zorovnik, J.: A distributed algorithm for bandwidth allocation in stable ad hoc networks. Technical Report 4827, INRIA (2003)
9. Gertz, M., Wright, S.: Object-oriented software for quadratic programming. Technical Report ANL/MCS-P891-1000, Argonne National Laboratory, Mathematics and Computer Science Division (2001)
10. Bonald, T., Massoulié, L.: Impact of Fairness on Internet Performance. In: Proceedings of SIGMETRICS, Cambridge, MA, USA (2001)
11. Bertsekas, D., Gallager, R.: Data Networks. Prentice Hall (1987)

Locally Optimal Scatternet Topologies for Bluetooth Ad Hoc Networks

Tommaso Melodia and Francesca Cuomo

INFOCOM Department, University of Rome “La Sapienza”
Via Eudossiana 18 00184, Rome, ITALY
{tommaso, cuomo}@net.infocom.uniroma1.it

Abstract. *Bluetooth* is a promising technology for personal/local area wireless communications. A Bluetooth *scatternet* is composed of overlapping *piconets*, each with a low number of devices sharing the same radio channel. This paper discusses the *scatternet formation* issue by analyzing topological characteristics of the scatternet formed. A matrix-based representation of the network topology is used to define metrics that are applied to evaluate the key cost parameters and the scatternet performance. Numerical examples are presented and discussed, highlighting the impact of metric selection on scatternet performance. Then, a distributed algorithm for *scatternet topology optimization* is introduced, that supports the formation of a “locally optimal” scatternet based on a selected metric. Numerical results obtained by adopting this distributed approach to optimize the network topology are shown to be close to the global optimum.

1 Introduction

*Bluetooth*¹ is a promising technology for ad hoc networking that could impact several wireless communication fields providing WPAN (*Wireless Personal Area Networks*) extensions of public radio networks (e.g., GPRS, UMTS, Internet) or of local area ones (e.g. 802.11 WLANs, Home RF) [1][2]. The Bluetooth system supports a 1 Mbit/s gross rate in a so-called *piconet*, where up to 8 devices can simultaneously be inter-connected. The radius of a piconet (*Transmission Range*, TR) is about 10 meters for Class 3 devices.

One of the key issues associated with the BT technology is the possibility of dynamically setting-up and tearing down piconets. Different piconets can coexist by sharing the spectrum with different frequency hopping sequences, and inter-connect in a scatternet. When all nodes are in radio visibility, scenario which we will refer to as *single hop*, the formation of overlapping piconets allows more than 8 nodes to contemporarily communicate and may enhance the system capacity. In a *multi-hop* scenario, where nodes are not all in radio vicinity, a scatternet is mandatory to develop a connected platform for ad-hoc networking.

¹ This work was partially funded by MIUR (Italian Ministry of Research) in the framework of the VICOM (Virtual Immersive COMMunications) project.

This paper addresses the scatternet formation issue by considering topological properties that affect the performance of the system. Most works in literature aim at forming a connected scatternet while performance related topological issues typically remain un-addressed. To this aim we introduce a matrix based scatternet representation that is used to define metrics and to evaluate the relevant performance. We then propose a distributed algorithm that performs topology optimization by relying on the previously introduced metrics. We conclude by describing a two-phases scatternet formation algorithm based on the optimization algorithm. To the best of our knowledge, this is the first scatternet formation algorithm explicitly aimed at optimizing the topology of the network.

The paper is organized as follows. Section 2 recalls the main aspects related to the piconet and scatternet models. Section 3 briefly summarizes the state of the art in scatternet formation, while in Section 4 a framework for scatternet analysis, based on a matrix representation is presented, together with simple metrics to evaluate scatternet performance. Section 5 presents the Distributed Scatternet Optimization Algorithm (DSOA) while Section 6 describes a two-phase scatternet formation algorithm based on DSOA. Section 7 concludes the paper.

2 Bluetooth Basics

Bluetooth exploits an 83.5 MHz band, divided into 79 equally spaced 1 MHz channels [1]. The multiple access technique is the FHSS-TDD (Frequency Hopping Spread Spectrum - Time Division Duplexing). Two Bluetooth units exchange information by means of a master-slave relationship. Master and slave roles are dynamic: the device that starts the communication acts as master, the other one as slave. After connection establishment, master and slave exchange data by hopping at a frequency of 1600 hops/second on the 79 available channels. Different hopping sequences are associated to different masters.

A master can connect with up to 7 slaves within a *piconet*. Devices belonging to the same piconet share a 1 Mbit/s radio channel and use the same frequency hopping sequence. Only communications between master and slaves are permitted. Time is slotted and the master, by means of a polling mechanism, centrally regulates the medium access. Thanks to the FHSS, which is robust against interference, multiple piconets can coexist in the same area. Considerable performance degradation only occurs for a high number of co-located piconets (in the order of 50) [3].

A *scatternet* is defined as an interconnection of overlapping piconets. Each device can join more than one piconet, and participate to communications in different piconets on a time-division basis. Devices that belong to more than one piconet are called *gateways* or *BridGing units* (BG).

Since there are many topological alternatives to form a scatternet out of the same group of devices, the way a scatternet is formed considerably affects its performance.

3 Related Work

Scatternet formation in Bluetooth has recently received a significant attention in the scientific literature. Existing works can be classified as single-hop [4][5][6][7] and multi-hop solutions [8][9][10][11][12].

Paper [4] addresses the Bluetooth scatternet formation with a distributed logic that selects a leader node which subsequently assigns roles to the other nodes in the system. In [5] a distributed formation protocol is defined, with the goal of reducing formation time and message complexity. In [5] and [6], the resulting scatternet has a number of piconets close to the theoretical minimum. The works in [7], [8] and [9] form tree shaped scatternets. In [7], Tan et al. present the TSF (Tree Scatternet Formation) protocol, which assures connectivity only in single-hop scenarios. Zaruba et al. propose a protocol which operates also in a multi-hop environment [8] but is based on time-outs that could affect the formation time. SHAPER [9] forms tree-shaped scatternets, works in a multi-hop setting, shows very limited formation time and assures self-healing properties of the network, i.e. nodes can enter and leave the network at any time without causing long term loss of connectivity.

A second class of multi-hop proposals is based on clustering schemes. These algorithms principally aim at forming connected scatternets. In [10] and [11] the *BlueStars* and *BlueMesh* protocols are described respectively. Also [12] defines a protocol that limits the number of slaves per master to 7 by applying the *Yao* degree reduction technique. The proposed algorithm assumes that each node knows its geographical position and that of each neighbor.

Recently, the work in [13] proposed a new on-demand route discovery and construction approach which, however, requires substantial modifications to the Bluetooth standard to guarantee acceptable route-setup delay.

Some other works discuss the optimization of the scatternet topology. This issue is faced in [14] and [15] by means of centralized approaches. In [14] the aim is minimizing the load of the most congested node in the network, while [15] discusses the impact of different metrics on the scatternet topology. In [16], an analytical model of a scatternet based on queuing theory is introduced, aimed at determining the number of non-gateway and gateway slaves to guarantee acceptable delay characteristics.

4 The Scatternet Formation Issue

Before addressing the issue of scatternet formation, we introduce a suitable scatternet representation.

4.1 Scatternet Representation

Let us consider a scenario with N devices. The scenario can be modelled as an undirected graph $G(V, E)$, where V is the set of nodes and an edge e_{ij} , between any two nodes v_i and v_j , belongs to the set E iff $distance(v_i, v_j) < TR$, i.e., if v_i and v_j are within each other's transmission range. $G(V, E)$ can be represented

by an $N \times N$ adjacency matrix $A = [a_{ij}]$, whose element a_{ij} equals 1 iff device j is in the TR of device i (i.e., j can directly receive the transmission of i).

Besides the *adjacency graph* $G(V, E)$, we model the scatternet with a *bipartite graph* $G_B(V_M, V_S, L)$, where $|V_M| = M$ is the number of masters, $|V_S| = S$ is the number of slaves, and L is the set of links (with $N = M + S, V_M \cap V_S = \{\emptyset\}, V_M \cup V_S = V$). A link may exist between two nodes only if they belong to the two different sets V_M and V_S . Obviously, for any feasible scatternet, we have $L \subseteq E$. This model is valid under the hypothesis that a master in a piconet does not assume the role of slave in another piconet; in other words, by adopting this model, the BGs are slaves in all the piconets they belong to. We rely on this hypothesis to slightly simplify the scatternet representation, the complexity in the description of the metrics and to reduce the space of possible topologies. Moreover, intuitively, the use of master/slave BGs can lead to losses in the system efficiency. If the BG is also a master, no communications can occur in the piconet where it plays the role of master when it communicates as slave. However, to the best of our knowledge, this claim has never been proved to be true. Future work will thus extend the results presented in this paper to non-bipartite graphs.

The bipartite graph G_B can be represented by a rectangular $M \times S$ binary matrix \mathbf{B} . In \mathbf{B} , each row is associated with one master and each column with one slave. Element b_{ij} in the matrix equals 1 iff slave j belongs to master i 's piconet. Moreover, a *path* between a pair of nodes (h, k) can be represented by another $M \times S$ matrix $\mathbf{P}^{h,k}(\mathbf{B})$, whose element $p_{ij}^{h,k}$ equals 1 iff the link between master i and slave j is part of the path between node h and node k ($1 \leq i, j, h, k \leq N$). To finish with, we will say that an $M \times S$ rectangular matrix \mathbf{B} represents a "Bluetooth-compliant" scatternet with M masters and S slaves if it represents a fully connected network (i.e., the matrix does not have a block structure, notwithstanding permutations of the rows), and no more than 7 slaves belong to each piconet (the sum of the elements of each row is less than 7).

4.2 Metrics for Scatternet Performance Evaluation

In [15], we introduced some metrics for scatternet evaluation. These metrics can either be dependent on or independent of the traffic loading the scatternet. For the convenience of the reader, we recall the Traffic Independent (TI) metrics which will be considered in the following.

A first traffic independent metric is the overall capacity of the scatternet. Evaluating such a capacity is not an easy task, since it is related to the capacity of the composing piconets which in turn depends on the intra-piconet and inter-piconet scheduling policies. To the best of our knowledge, no such evaluation is available in literature. In the following, we introduce a simple model to estimate the capacity of a scatternet and we exploit this evaluation for scatternet formation. In the model we assume that:

- a master may offer the same amount of capacity to each of its slaves by equally partitioning the piconet capacity;
- a BG slave spends the same time in any piconet it belongs to.

These assumptions are tied to intra and inter piconet scheduling; here, for the sake of simplicity, we assume policies that equally divide resources; however the model can be straightforwardly extended to whatever scheduling policy.

The scatternet capacity will be evaluated by normalizing its value to the overall capacity of a piconet (i.e., 1 Mbit/s). Let us define two $M \times S$ matrices, $\mathbf{O}_{\mathbf{TI}}(\mathbf{B}) = [o_{ij}]$, and $\mathbf{R}_{\mathbf{TI}}(\mathbf{B}) = [r_{ij}]$ with $o_{ij} = b_{ij}/s_i$ and $r_{ij} = b_{ij}/m_j$, where s_i denotes the number of slaves connected to master i and m_j denotes the number of masters connected to slave j (for $j = 1, \dots, S$ and $i = 1, \dots, M$):

$$m_j = \sum_{i=1}^M b_{ij}, j = 1, \dots, S \quad s_i = \sum_{j=1}^S b_{ij}, i = 1, \dots, M \quad (1)$$

The matrix $\mathbf{O}_{\mathbf{TI}}(\mathbf{B})$ represents the portions of capacity a master may offer to each of its slaves. The $\mathbf{R}_{\mathbf{TI}}(\mathbf{B})$ matrix represents the portions of capacity a slave may "spend" in the piconet it is connected to. The overall capacity of the scatternet is given by the sum of the capacities of all links. The capacity c_{ij} of link (i, j) is the minimum between the capacity o_{ij} and the capacity r_{ij} . Let us define the matrix $\mathbf{C}_{\mathbf{TI}}(\mathbf{B})$, whose elements represent the normalized link capacity, as:

$$\mathbf{C}_{\mathbf{TI}}(\mathbf{B}) = [c_{ij}] = [\min(o_{ij}, r_{ij})] \quad (2)$$

The associated metric is the *normalized capacity* $c_{TI}(\mathbf{B})$ of a scatternet defined as:

$$c_{TI}(\mathbf{B}) = \sum_{i=1}^M \sum_{j=1}^S \min(o_{ij}, r_{ij}) \quad (3)$$

As shown in [15], path lengths have a considerable impact on scatternet performance. As a consequence we introduce two metrics that do take into account path lengths. Let us denote, for a scatternet represented by a matrix \mathbf{B} , the length of the path between device h and device k (expressed in number of hops) as:

$$q^{h,k}(\mathbf{B}) = \sum_{i=1}^M \sum_{j=1}^S p_{ij}^{h,k} \quad (4)$$

We can now introduce the *average path length*, which is the path length averaged over all possible source-destination couples, and is given by:

$$q_{TI}(\mathbf{B}) = \sum_{h=1}^N \sum_{k=1, k \neq h}^N \frac{q^{h,k}(\mathbf{B})}{N \cdot (N - 1)} \quad (5)$$

Obviously, we want $q_{TI}(\mathbf{B})$ to be minimized.

Given the capacity of a scatternet $c_{TI}(\mathbf{B})$ and the relevant average path length $q_{TI}(\mathbf{B})$, the capacity available, on average, for the generic source-destination couple among the nodes in \mathbf{B} is given by:

$$a_{TI}(\mathbf{B}) = \frac{c_{TI}(\mathbf{B})}{q_{TI}(\mathbf{B}) \cdot N \cdot (N - 1)} \quad (6)$$

This last metric, which we will refer to as *average path capacity*, will be considered in all the experiments reported in the following. As we showed in [15], scatternets with high values of this metric show a good compromise between capacity and path length.

5 A Distributed Algorithm for Topology Optimization

In this section we describe a Distributed Scatternet Optimization Algorithm (DSOA), that aims at optimizing the topology to obtain a performance (in terms of the chosen metric) as close as possible to the optimum. Note that the selection of the optimized topology is decoupled from the establishment of the links that compose it, as will become clearer in Section 6, where we will describe a two-phases distributed scatternet formation algorithm based on DSOA.

5.1 Distributed Scatternet Optimization Algorithm (DSOA)

We consider the adjacency graph $G(V, E)$. First, we aim at obtaining an ordered set of the nodes in V . The first procedure orders the nodes in the graph according to a simple property: a node k must be in transmission range of at least one node in the set $1..k - 1$.

Procedure 1 ORDER_NODES

Input: $G(V, E)$
Output: ordered set of the nodes in V , $W = \{w_k\}$, $k = 1, 2, \dots, N$, $N = |V|$
begin
 w_1 = random selection of a node v from V
 $W = w_1$
for $k = 2 : N$ **do**
 w_k = random selection of a node v from V such that:
 1. $v \notin W$
 2. $\exists u \in W$ such that $distance(u, v) \leq TR$
 $W = W \cup w_k$
end for
end

Since with DSOA the nodes sequentially select how to connect, each node must be in TR of at least another node already entered. The following proofs that it is always possible to obtain such an ordering of the nodes, i.e. that this procedure always ends.

Theorem 1 *Given a connected graph $G(V, E)$, the procedure ORDER_NODES always terminates, and $|W| = N$.*

Proof: Suppose that at some step k of the procedure, $k < N$, we have $W = \{w_1, w_2, \dots, w_{k-1}\}$ and no couple (v, w) with $v \in (V \setminus W)$, $w \in W$ exists such that

$distance(v, w) < TR$. Therefore, since $W \subseteq V$, there exist two disconnected components W and $V \setminus W$ of $G(V, E)$.

At the end of this procedure, then, node k is in transmission range of at least one of the nodes $1, 2, \dots, k-1$. The second procedure is the core of the algorithm. Here we let e_{ij} be the link between the nodes w_i and w_j of a scatternet ($1 \leq i, j \leq N$). This part of the algorithm is also dependent on the selected metric M . At each step k , node w_k “enters” in the scatternet in the best possible way, according to M .

Procedure 2 SCATTERNET_OPTIMIZATION_ALGORITHM

Input: $W, G(V, E), M$

Output: Locally Optimal Scatternet \mathbf{B}^*

begin

$V_M = \emptyset$

$V_S = \emptyset$

$V_M = V_M \cup w_1$

$V_S = V_S \cup w_2$

$\mathbf{B}^2 = [1]$

for $k = 3 : N$ **do**

case 1: consider w_k in V_M

 * derive all Bluetooth-compliant matrices \mathbf{B}^k with $|V_M| + 1$ rows and $|V_S|$ columns
 calculate values of $M(\mathbf{B}^k)$

case 2: consider w_k in V_S

 * derive all Bluetooth-compliant matrices \mathbf{B}^k with $|V_M|$ rows and $|V_S| + 1$ columns
 calculate values of $M(\mathbf{B}^k)$

 select the \mathbf{B}^k with optimal $M(\mathbf{B}^k)$

if optimum in *case 1* **then**

$V_M = V_M \cup w_k$

else

if optimum in *case 2* **then**

$V_S = V_S \cup w_k$

else

 RECONFIGURE($\mathbf{B}^{k-1}, V_M, V_S$)

end if

end if

end for

$\mathbf{B}^* = \mathbf{B}^N$

end

The RECONFIGURE procedure is executed in the unlikely case when w_k is only in transmission range of master nodes that have already 7 slaves in their piconet. For the sake of simplicity, details of this procedure are only given in the following proof of correctness. In this case, one of the 7 slaves is forced to become master of one of the other slaves. This is shown to be always possible. The following proves the correctness of SOA, i.e. it is always possible for a node to enter the network respecting the Bluetooth properties.

Proof of correctness. Node w_2 is in transmission range of w_1 , thus the two nodes can connect. Each node w_k , with $k > 2$ can always establish a new piconet, thus connecting as a master, whenever a node $v \in \{w_1, w_2, \dots, w_{k-1}\}$ exists s.t. $v \in V_S$ and $distance(w_k, v) \leq TR$, i.e. one of the slave nodes already in the network is in transmission range of w_k . If no slaves are in TR of w_k , whenever a node $v \in V_M$ exists, with $distance(w_k, v) \leq TR$, and $slaves(v) \leq 7$, w_k can be a slave of v . Otherwise, at least one node $w_i \in V_M$ must exist, with $distance(w_k, v) \leq TR$, and $slaves(v) = 7$, with $i \leq k$. The RECONFIGURE procedure can always be executed in this way. If at step i node w_i selected more than 1 slave, it can disconnect from the slave that causes the minimum decrease/increase in the metric value. The topology is still connected, and w_k can select w_i as its slave. If, otherwise, w_i selected only one slave at step i , this cannot be disconnected, since this could cause loss of connectivity for the network. Thus, one of the other 6 slaves must be disconnected. However, it was proven in [8] that in a piconet with at least 5 slaves, at least 2 of them are in TR of each other. Thus, at least one of the slaves can become master and select another slave. The network can therefore be reconfigured by forcing the 7th slave that connected to w_i to become master of another slave of w_i , to minimize reconfigurations. If it is not in TR of any other slave of w_i , we can try with the 6th, and so on. At least one of the six slaves must be able to become master and select one of the other 5 as its slave.

The local optimization in SOA (steps with mark *) can be performed by means of state space enumeration, as in the simulations results we show, or, e.g., by means of randomized local search algorithms.

The distributed version of the SOA (Distributed SOA, DSOA) straightforwardly follows. At each step k , a new node w_k receives information on the topology selected up to that step (\mathbf{B}^{k-1} matrix) and selects the role (master or slave) it will assume and the links it will establish, with the aim of maximizing the global scatternet metric. If the node becomes a master it will select a subset of the slaves in its TR already in the scatternet; if it becomes a slave it will select a subset of the masters in its TR , already in the scatternet. ORDER_NODES is needed to guarantee that, when node k enters, it can connect to at least one of the previously entered nodes. DSOA can be classified as a *greedy* algorithm, since it tries to achieve the optimal solution by selecting at each step the *locally optimal* solution, i.e. the solution that maximizes the metric of the overall scatternet, given local knowledge and sequential decisions. Greedy algorithms do not always yield the global optimal solution. As will be shown in the next subsection, however, the results obtained with DSOA are close to the optimum.

5.2 Examples and Numerical Results (DSOA)

In this section we show some results obtained with DSOA, by using average path capacity as a metric. As previously discussed, we believe that average path capacity is a good metric since it takes into account both capacity and average path length of the scatternet. Figure 1 shows a comparison between the optimal $a_{TI}(\mathbf{B})$ and the one obtained with the DSOA. The dotted curve shows the

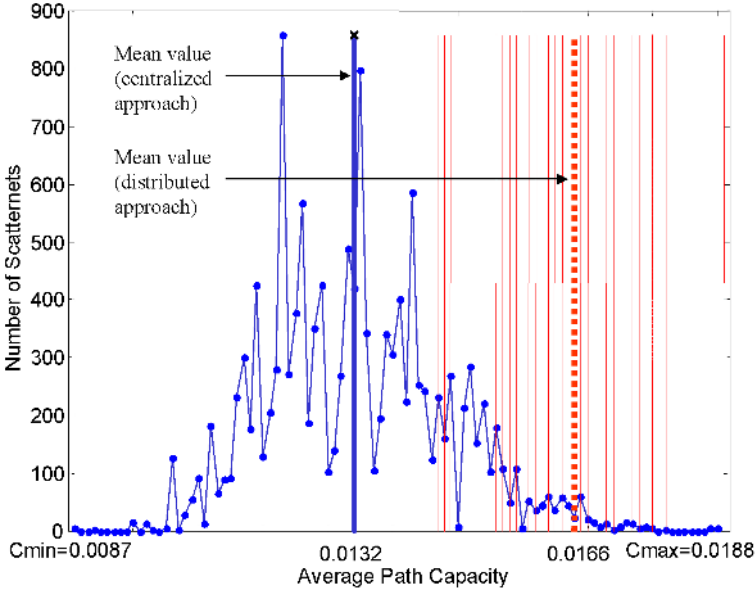


Fig. 1. Results obtained with DSOA compared with the distribution of the metric values

histogram of the average path capacity of all possible Bluetooth-compliant scatternets feasible in this scenario. The histogram of the average path capacity of all the feasible scatternets in a scenario constituted by 10 nodes distributed in an area of 25x25 meters is represented. As will be shown later, a similar distribution holds in general. It is easy to see that, already with 10 nodes, the number of different feasible topologies is very high. The values of $a_{TI}(\mathbf{B})$ are distributed in a range starting from $a_{TI,min}(\mathbf{B}) = 0.0087$ (≈ 8 kbit/s for every possible node pair) to $a_{TI,max}(\mathbf{B}) = 0.0188$ (≈ 19 kbit/s per pair); the mean value of $a_{TI}(\mathbf{B})$ is also shown (equal to 0.0132). Note that the mean value is quite distant from the maximum value, which corresponds to the value associated with the optimal scatternet. Moreover, a few scatternets have a high value of $a_{TI}(\mathbf{B})$ and are thus contained in the right tail of the histogram. This is an interesting result because it suggests that topology optimization is a fundamental issue for Bluetooth scatternets: in fact, this distribution of the metric values means that it is highly unlikely to obtain a high performance scatternet by randomly selecting a topology. We need to deploy protocols that not only search for a connected scatternet but also explicitly aim at maximizing its performance. As regards the DSOA, the vertical lines in Figure 1 correspond to the values of $a_{TI}(\mathbf{B})$ for 100 different scatternets formed by using 100 different randomly chosen sequential orders. The lines are concentrated in the right part of the figure (i.e., the scatternets formed have a value of $a_{TI}(\mathbf{B})$ greater than the overall mean value of all possible scatternets). The mean value of $a_{TI}(\mathbf{B})$ of these 100 DSOA scatternets is equal to 0.0166. The un-normalized values of the average capacity per path

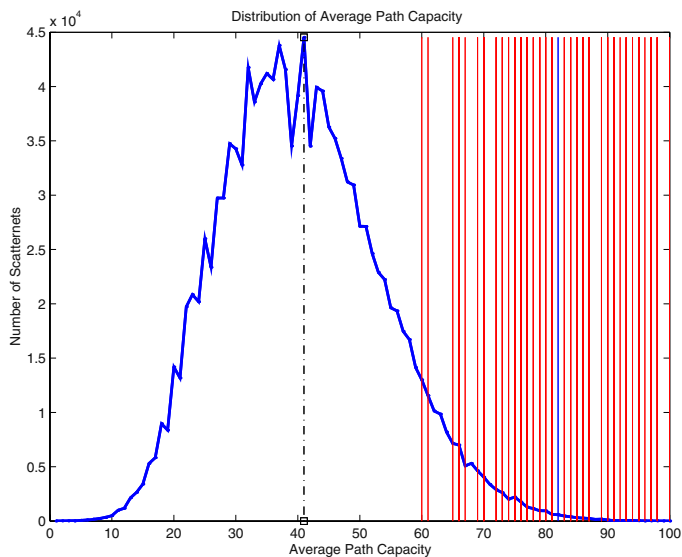


Fig. 2. Distribution of Average Path Capacity for 15 nodes

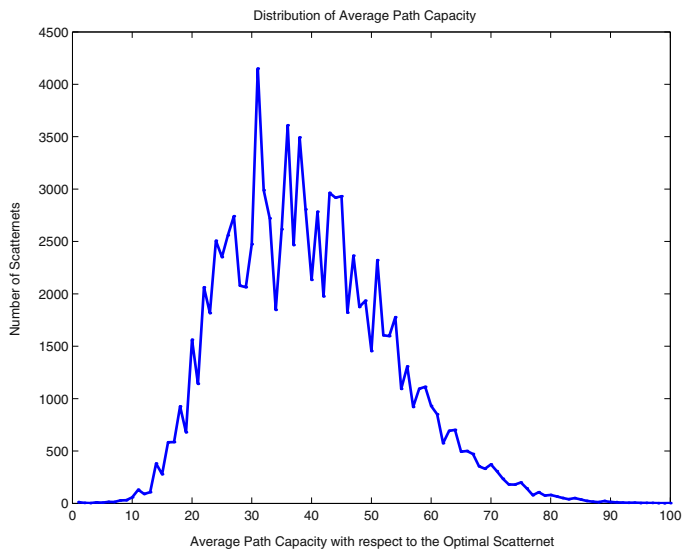


Fig. 3. Distribution of average path capacity on different scenarios

obtained with DSOA is about 17 kbit/s, while the maximum possible value is 19 kbit/s; this confirms the good behavior of the DSOA.

Figure 2 shows a similar distribution in a scenario with 15 nodes in a multi-hop context. In Figure 3 a distribution averaged over 100 different scenarios, with varying number of nodes, is shown; Figure 4 reports the distribution of the

values obtained with DSOA in the same scenarios. The probability of obtaining a value of the metric between the optimal and 70% of the optimal by randomly selecting a topology is very low; by using DSOA this probability is close to 1. For a higher number of nodes the state-space enumeration approach, which has been useful in obtaining the distribution of the metric values, becomes unfeasible. The conclusion we can draw from the above figures is that scatternets formed with DSOA have a structure quite similar to the optimal ones, obtained with the centralized approach. Correspondingly, the value of the metric obtained with DSOA is close (sometimes equal) to the one obtained with the centralized approach. The same behavior has been observed in numerous experiments, carried out with different metrics and number of nodes.

6 A Two-Phases Scatternet Formation Algorithm

The actual Distributed Scatternet Formation Protocol is divided in two phases:

1. Tree Scatternet Formation (SHAPER);
2. DSOA and new Connections Establishment.

To implement DSOA we need a mechanism to distribute the “right” to enter in the network to every node k at step k , and to convey the topology selected by the previous $k - 1$ nodes (\mathbf{B}^{k-1} matrix). The distributed implementation in Bluetooth however is not simple since the system lacks a shared broadcast medium that would allow signaling among nodes. A good solution which guarantees: i) the required ordering of the nodes; ii) synchronization of the decisions; iii) a shared communication medium, is to form a tree-shaped “provisional” scatternet. A tree-shaped scatternet can asynchronously be formed in a distributed fashion. In [9], we proposed a new protocol for tree scatternets (SHAPER), which works in an asynchronous and totally distributed fashion, thus allowing the self-organized formation of a tree shaped scatternet in a multi-hop context. We showed that a tree scatternet can be formed in a few seconds time, and that less time is required when nodes are denser.

After the tree has formed, a simple recursive visit procedure can be executed on it, which allows implementing the DSOA topology optimization process. It is easy to see that a sequential visit of all nodes in the tree, from the root down to the leaves, guarantees the order provided by ORDER_NODES. We let $parent(v)$ be the parent of v in the tree and $children(v)$ be the set of children nodes for v . Step k of the distributed procedure is executed on a node when it receives an EXECUTE_ENTER(\mathbf{B}^{k-1}, k) message from its parent. \mathbf{B}^{k-1} is the matrix representing the topology selected by the previously visited nodes. The root node resulting from SHAPER starts the distributed execution of such procedure at the expiration of a timeout.

When a given node v starts the ENTER procedure, it executes the DSOA, i.e. it decides how to enter in the network. Then, the node randomly picks up one of its children nodes, and sends the EXECUTE_ENTER($\mathbf{B}^k, k + 1$) message to it. This causes the execution of the ENTER procedure on the child. After sending the EXECUTE_ENTER command, v waits for an answer message (BRANCH_ENTERED)

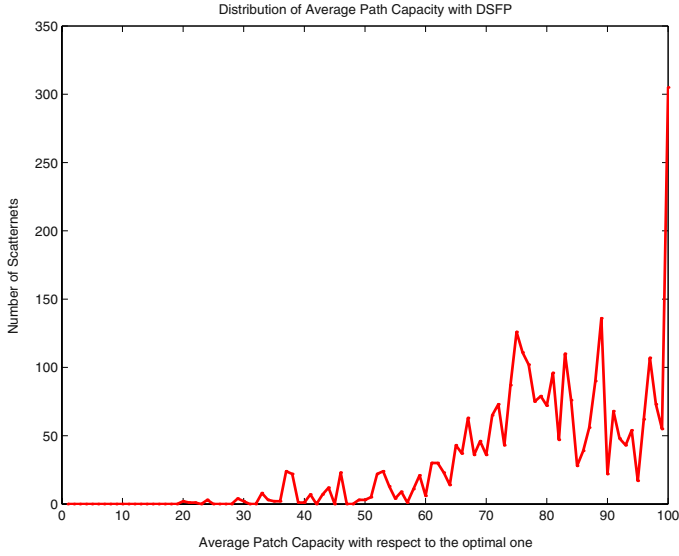


Fig. 4. Distribution of average path capacity for DSOA scatternets

from the child. This contains information about the topology selected by the whole *branch* which goes down from v to the *leaf* nodes. After the answer from the child is received, v selects another child and does the same. When v receives the answer from its last child, it informs its parent of the topology selected by itself and by all of its descendants with the `BRANCH_ENTERED` message. When the *root* node receives the answer from its last son, all nodes have taken their decision.

The last step concerns the actual connection establishment. The root node broadcasts the matrix representing the final scatternet structure. The matrix is recursively broadcasted at every level of the tree. Once a node has broadcasted the matrix down to its children in the tree, it enters the *topology reconfiguration*

Procedure 3 ENTER

```

begin
   $\mathbf{B}^k = \text{DSOA}(\mathbf{B}^{k-1})$ 
  for each  $v \in \text{children}$  do
    send(EXECUTE_ENTER( $\mathbf{B}^k, k + 1$ ),  $v$ )
    wait_answer()
    [ $\mathbf{B}^{k+c}, c$ ] = answer( $v$ )
     $k = k + c$ 
  end for
  send(BRANCH_ENTERED( $\mathbf{B}^k, k$ ), parent)
end

```

phase. During this phase the node can start establishing the connections that will compose the optimized scatternet. Every link that is not already part of the tree topology has to be established. Redundant links have to be torn down. Every node alternates between a *communication* and a *formation* state. During the latter the node tries to establish the new links, while during the former user data is transmitted so as to guarantee the continuity of service during the reconfiguration phase. If a node has a master role in the optimized scatternet, it *pages* its first slave. When the connection is established, it continues with the other ones. If the node has a slave role, it will *page scan* for incoming connections. Priority is given to previously entered masters so as to avoid deadlocks. Every node starts tearing down the old links only when the new ones have been established, so as to preserve connectivity. Since all nodes know the overall topology, the routing task is also simplified. Route discovery algorithms have to be implemented only when mobility has to be dealt with or for other particular situations.

The most time consuming phase of the algorithm is the formation of the tree, which, as said before, becomes necessary because Bluetooth lacks a shared broadcast medium. However, in [9] we showed that the tree can be formed in a few seconds. During the tree formation phase data exchange among nodes can start, so users don't have to wait for the overall structure to be set up. Data exchange can continue on the provisional tree scatternet during the optimization process. Work is in progress to add self-healing functionalities to the algorithm (nodes can enter and exit the network which is re-optimized periodically) and to simulate the integration of SHAPER and DSOA.

7 Conclusions

In this paper, the scatternet formation issue in Bluetooth was discussed, by setting a framework for scatternet analysis based on a matrix representation, which allows developing and applying different metrics. A distributed algorithm for Scatternet Topology Optimization, DSOA, was described. The performance of DSOA was evaluated and shown to be encouraging: the distributed approach gives results very similar to a centralized one. The integration with the SHAPER Scatternet Formation Algorithm and other implementation concerns have been discussed. Ongoing activities include the full design of a distributed scatternet formation algorithm which implements DSOA and deals with mobility and failures of nodes, as well as a simulative evaluation of the time needed to set-up a scatternet and its performance in presence of different traffic patterns.

References

1. J. Haartsen, "The Bluetooth Radio System", *IEEE Personal Communications*, Vol. 7, n. 1, pp. 28–36, February 2000.
2. P. Johansson, R. Kapoor, M. Gerla, M. Kazantzidis, "Bluetooth an Enabler of Personal Area Networking", *IEEE Network*, Special Issue on Personal Area Networks, pp. 28–37, September/October 2001.

3. S. Zurbes, "Considerations on Link and System Throughput of Bluetooth Networks", Proc. of the *11th IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Vol. 2, pp. 1315–1319, 2000.
4. T. Salonidis, P. Bhagwat, L. Tassiulas, R. La Maire, "Distributed topology construction of Bluetooth personal area networks", Proc. of the *IEEE Infocom 2001*, pp. 1577–1586, April 2001.
5. C. Law, A. Mehta, K-Y Siu, "Performance of a new Bluetooth scatternet formation protocol", Proc. of the *Mobihoc 2001*, pp. 183–192, 2001.
6. H. Zhang, J. C. Hou, L. Sha, "A Bluetooth Loop Scatternet Formation Algorithm" Proc. of the *IEEE International Conference on Communications (ICC 2003)*, pp. 1174–1180, May 2003.
7. G. Tan, A. Miu, J. Gutttag, H. Balakrishnan, "An Efficient Scatternet Formation Algorithm for Dynamic Environments", in *IASTED Communications and Computer Networks (CCN)*, Cambridge, November 2002.
8. G. Zaruba, S. Basagni, I. Chlamtac, "Bluetrees – Scatternet formation to enable Bluetooth-based personal area networks", Proc. of the *IEEE International Conference on Communications (ICC 2001)*, pp. 273–277, 2001.
9. F. Cuomo, G. Di Bacco, T. Melodia, "SHAPER: a Self Healing Algorithm Producing multihop bluetooth scatterNets", Proc. of the *IEEE Globecom 2003*, San Francisco, December 2003.
10. C. Petrioli, S. Basagni, I. Chlamtac "Configuring BlueStars: multihop scatternet formation for Bluetooth networks", *IEEE Transactions on Computers*, Vol. 52, Issue 6, pp. 779–790, June 2003.
11. C. Petrioli, S. Basagni, I. Chlamtac, "BlueMesh: Degree-constrained multihop scatternet formation for Bluetooth networks", *ACM/Kluwer Journal on Special Topics in Mobile Networking and Applications (MONET)*, Special Issue on Advances in Research of Wireless Personal Area Networking and Bluetooth Enabled Networks, 2002.
12. I. Stojmenovic, "Dominating set based scatternet formation with localized maintenance", Proc. of the *Workshop on Advances in Parallel and Distributed Computational Models*, April 2002.
13. Y. Liu, M. J. Lee, T. N. Saadawi, "A Bluetooth Scatternet-route Structure for Multihop Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 21, n. 2, pp. 229–239, February 2003.
14. M. Ajmone Marsan, C. F. Chiasserini, A. Nucci, G. Carello, L. De Giovanni, "Optimizing the Topology of Bluetooth Wireless Personal Area Networks", *IEEE INFOCOM 2002*, Vol.2, pp. 572–579, June 2002.
15. F. Cuomo, T. Melodia, "A General Methodology and Key Metrics for Scatternet Formation in Bluetooth", Proc. of the *IEEE Globecom 2002*, Taipei, November 2002.
16. R. Kapoor, M. Y. M. Sanadidi, M. Gerla, "An Analysis of Bluetooth Scatternet Topologies," Proc. of the *IEEE International Conference on Communications (ICC 2003)*, pp. 266–270, 2003.

An On-Demand Bluetooth Scatternet Formation Algorithm^{*}

Elena Pagani, Gian Paolo Rossi, and Stefano Tebaldi

Computer Science Dept., Università degli Studi di Milano
{pagani,rossi}@dico.unimi.it, Stefano.Tebaldi@unimi.it

Abstract. In this paper, we propose the *On-Demand Bluetooth scatternet formation algorithm* (ODBT). ODBT characterizes an ad hoc infrastructure with a tree topology. It is able to cope with topology changes due to either leaving or moving Bluetooth devices, as well as with devices that dynamically join the scatternet. It can support out-of-range devices. We describe in detail how ODBT can be implemented in the Bluetooth protocol stack, and we analyze its performance in comparison with other proposals existing in the literature, also by means of simulation techniques.

1 Introduction

The Bluetooth technology has been designed with the purpose of replacing cabling amongst neighbor devices, for instance to connect computers and mobile phones to external devices and accessories via wireless links. Bluetooth is an interesting solution to rapidly deploy wireless infrastructures using low cost, easily available devices such as PDAs, notebooks and cellular phones.

The base Bluetooth network infrastructure is represented by a *piconet*, that is formed by up to 8 Bluetooth devices (BDs) actively participating in the communications, one of which has the role of *master* while the others act as *slaves*. The communication range of a BD is around 10-30 mt., arriving for some devices to up to 100 mt.; this range is the maximum piconet radius. The Bluetooth specification includes the possibility of building *scatternets*, obtained by connecting several piconets into an ad hoc infrastructure. The scatternets allow to increase the communication range and the number of BDs involved in a system. Yet, the scatternet formation mechanism is not provided by the specifications, and is currently matter of research.

In this paper, we present the *On-Demand Bluetooth Scatternet Formation algorithm* (ODBT) to characterize a communication infrastructure connecting a set of BDs in a tree topology. The scatternet formation is started when needed by an initiator node, that becomes the tree root; the other BDs are progressively grafted to the structure, so that the overall topology is optimized with respect to the latency in the data forwarding. Typically, the tree root can be the group

^{*} This work has been partially supported by the Italian Ministry of Education, University and Research in the framework of the FIRB “Web-Minds” project.

coordinator or the data source. The tree structure guarantees the absence of loops and minimizes the number of roles each BD can have in the scatternet, thus reducing the probability that a BD can represent a bottleneck.

2 Bluetooth Architecture

In this section we provide a brief overview of the Bluetooth protocol stack and operations. The complete Bluetooth specification is provided in [2]. In figure 1(a), we show the low layers of the Bluetooth protocol stack. As the wireless media, Bluetooth exploits the internationally unlicensed ISM band ranging from 2.4 GHz to 2.48 GHz. It adopts Frequency Hopping Spread Spectrum as the radio transmission technique, to achieve robustness to the interferences. The hop rate is 1600 hops per second; hence, each hop slot length is $625 \mu\text{sec}$. The channel is partitioned in 79 subchannels having 1 MHz bandwidth each. The baseband services are exploited by the Bluetooth devices (BDs) to agree on the frequency hop sequence and to establish the links. The functionalities for the piconet set-up are involved in the baseband and the Link Manager Protocol (LMP) layers. LMP also allows to perform the BDs, links and packets configuration. The baseband and LMP services are accessed through the Host Controller Interface (HCI). The

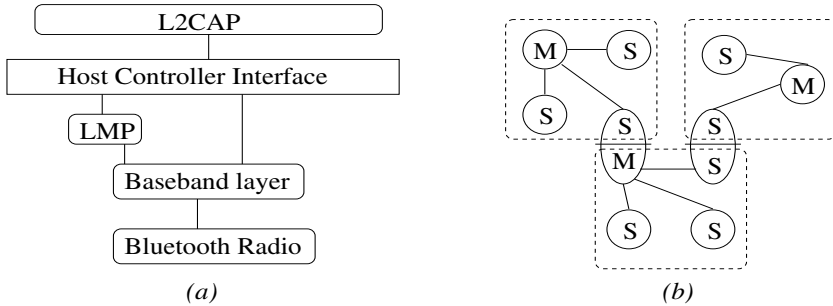


Fig. 1. (a) Bluetooth architecture. (b) Example of a scatternet infrastructure.

network layer services are provided by the Logical Link Control and Adaptation Protocol (L2CAP).

The piconet is built in two steps: the **inquiry** phase and the **page** phase. In the former step, a BD may discover its neighbors by sending broadcast messages (**inquiry** procedure). A BD that wants to be discovered performs the **inquiry_scan** procedure: upon receiving an **inquiry** message, it replies with its own 48-bits MAC layer address (BD_ADDR) and an estimate of its clock value. In this phase, the BDs are not yet synchronized: in order to increase the probability that two BDs contemporarily use the same frequency and can then successfully communicate, an *inquiry hopping sequence* is used involving 32 frequencies. Those frequencies are split into two groups of 16 frequencies each (a

train); each train is repeatedly tested before switching to the other train. The BD_ADDRs and clock estimates of the discovered BDs are used in the successive **page** phase to establish a connection. The paged device (the slave) performs the **page_scan** procedure by listening for a hop pattern computed basing on its own BD_ADDR, that was previously sent to the master. The paging device (the master) enters the **page** state, in which it sends to the slave information concerning its own BD_ADDR and clock value, which is used to compute the frequency hopping pattern for the subsequent communications. Since the two BDs clocks are not yet synchronized, the **page** procedure is carried out similarly to the **inquiry** procedure: the paging pattern involves two trains of 16 frequencies each, that are repeatedly tested. The **page** procedure is carried out separately for each master's neighbor.

Once the channel has been successfully established at the end of the **page** procedure, the master and its slaves communicate exploiting a time-division duplex policy. The even-numbered slots are for master-to-slave communications; the odd-numbered slots are for slave-to-master communications. A slave can use a slot only if it is allowed to by the master in the immediately preceding slot.

As we said before, a master can manage up to 7 *active* slaves, that can participate in the communication; in the **page** phase, the master assigns to each active slave a 3-bits active member address (AM_ADDR). Indeed, the number of slaves grafted to a given master can be higher than 7, but the remaining slaves must be in *park* mode. A BD that does not need to participate in the communication can enter the park mode: the master substitutes the slave AM_ADDR with a PM_ADDR, that can be used to un-park the BD. A BD can enter the sniff mode to save energy: it listens to the channel only every T_{sniff} slots and it maintains its AM_ADDR. A slave can be put in hold mode by the master for a hold time *holdTO*; in the meanwhile, it does not listen to the channel, but, it can be active in other piconets and it maintains its AM_ADDR. When *holdTO* expires, the slave must wait for a re-synchronization message from the master.

When two piconets are connected into a scatternet structure, the BDs belonging to both piconets assume the role of *bridges*. In fig.1(b), three piconets are shown (dashed squares): two piconets are connected through a BD behaving as slave in both of them (S/S bridge); two piconets are connected through a BD behaving as slave in one piconet and as master in the other piconet (M/S bridge). The bridges take in charge the traffic forwarding amongst piconets. Since a BD can be active only in one piconet at a time, the bridges must become active alternatively in each piconet they are connected to. As a consequence, the bridges represent the potential bottlenecks.

2.1 Related Works

A few other works exist in the literature, that propose mechanisms to form scatternet infrastructures. In Bluetree [8], the scatternet formation is initiated by the Blueroot, whose identity is predetermined. The Blueroot starts connecting all its neighbors as slaves. Then, the Blueroot slaves act as masters for their neighbors and so on recursively. The maximum allowed number of slaves for each master is

5. To fulfil this constraint, in the second phase of the algorithm the scatternet is reconfigured by splitting the piconets that are too dense. Bluenet [7] builds scatternets by initially aggregating the BDs into piconets involving at most N_{max} slaves, and then by interconnecting the piconets. In the simulations presented in [7], N_{max} was equal to 5. In Bluenet, a bridge node must avoid to form multiple links with the same piconet. This is achieved by having the BDs that provide information about the piconet they currently belong to, when they are paged. BTCP [5] is a 3-phases algorithm: in the first phase an election protocol is carried out, at the end of which the chosen coordinator knows the identities of all the participants. In the second phase the coordinator chooses other $P - 1$ masters to form P piconets, and $P \cdot (P - 1)/2$ bridges, thus characterizing a completely connected topology among the piconets in the third phase. The probability that the scatternet is connected depends on the time spent in electing the coordinator. In [5] some considerations are presented, concerning the timers sizing. LMS [3] is a distributed randomized algorithm. The BDs try to aggregate into piconets, which are then merged to form a scatternet. Piconets are reconfigured to make them dense (by merging low populated piconets), thus minimizing the number of piconets composing the scatternet. It has been proved that the formed scatternets have a $O(\log n)$ diameter, with n the number of involved BDs, and the message complexity equals $O(n)$. Some extensions are outlined to support moving, joining and leaving nodes, as well as BDs that are not all in range with each other. TSF [6] characterizes a tree scatternet by building a forest of piconets and scatternets that are then merged. The merging procedure is such that it guarantees loop freeness. TSF is the unique solution that explicitly allows nodes to start communicating while the scatternet is under construction. Among the considered algorithms, only TSF explicitly addresses the problems related with the management of the node mobility, and BDs dynamically joining and leaving the scatternet. Anyway, TSF may fail in characterizing a connected scatternet if the nodes are not all in range of each other; in particular, it is not able to merge two scatternets if their roots are out of range.

The ODBT algorithm we propose in this work represents an improvement on the Bluetree algorithm. We study in depth the issues involved with the practical implementation of the scatternet formation algorithm in the Bluetooth architecture, and we describe mechanisms to support mobility and BDs dynamically joining and leaving the scatternet. ODBT can, to some extent, support out-of-range BDs; moreover, it allows the data forwarding during the tree construction. ODBT achieves a trade-off between the data transmission latency and the control overhead, trying to maintain a low tree depth. In sec. 4, we compare the above algorithms with ODBT.

3 Scatternet Formation Algorithm

In this section, we describe ODBT. For the sake of simplicity, in section 3.1 we firstly assume that the BDs are all in range, and that the BDs do not move, neither they dynamically join or leave the system; crash failures are not consid-

ered. In section 3.2, we discuss in detail the synchronization issues that allow the link maintenance and the message forwarding, by appropriately controlling the behavior of the bridge BDs switching between two piconets. In section 3.3, we finally relax the assumptions above by considering the mechanisms adopted to deal with dynamically changing membership and topology.

3.1 Algorithm Overview

The scatternet formation is initiated by the *scatternet root* (SR). We assume that each BD knows whether it has to assume the role of SR or not. This assumption is appropriate, as for the most part of the customized applications requiring a scatternet infrastructure, the BD coordinating the group activities is known at configuration time.

The scatternet is built trying to minimize the number of piconets by maximizing the number of BDs involved in each piconet, with a twofold purpose: the lower is the number of piconets, the lower are both the path lengths and the number of bridges, which are bottlenecks in the data forwarding because they have to switch between two piconets. To build the first scatternet layer, the SR performs the *inquiry* procedure, while all the BDs different from SR start performing the *inquiry_scan* procedure. The *inquiry* and *page* procedures are performed as usual to form a piconet, but they are repeated until the SR connects exactly 7 slaves¹. At that point, the SR notifies its slaves that they may in turn continue the scatternet formation, by becoming masters and connecting other BDs. This procedure is recursively repeated, with each master that, after having connected 7 slaves, authorizes them to become masters and to form new piconets. Notice that, continuing the scatternet formation only when a piconet is full does *not* guarantee that the obtained tree is balanced. Indeed, one branch may grow faster than another, because it is not required that a whole tree level is completed before starting a new one. Yet, this policy aims at filling up each piconet as much as possible before creating new piconets, while a good balancing is guaranteed by the contemporary start of all the bridges belonging to the same piconet. A perfectly balanced tree could be obtained only by having the SR that coordinates the tree growth level by level. This would be expensive in terms of the number of control messages that should be exchanged.

Once a BD is connected to a master, it does not reply to the *inquiries* of other BDs; as a consequence, all the bridges in the scatternet are of type master/slave and each BD belongs to at most two piconets. This also guarantees that the tree is loop free: indeed, a loop could form if a master connects as slave a BD already connected to another piconet.

The bridge BDs must alternate between the slave role in the upstream piconet and the master role in the downstream piconet. According to the Bluetooth specification, a link is considered broken if it is not used for a *supervisionTO* timer, whose default value is 20 sec. To guarantee that links are not disrupted in a piconet while a BD is active in the other, we choose to exploit the *Hold*

¹ Remind that we are assuming that all the BDs are in range.

Mode for two reasons: the hold mode does not require to reassign the BD active member address, and a BD in hold mode does not listen to the channel until the *holdTO* expires, thus allowing energy saving. As we discuss in sec.3.2, the *holdTO* value must be negotiated depending on the state of the downstream piconets. According to the Bluetooth specification, the hold mode negotiation requires one slot for the master to broadcast the `LMP_hold_request` and one slot for each slave to reply with a `LMP_hold_accepted`. In figure 2, we show the

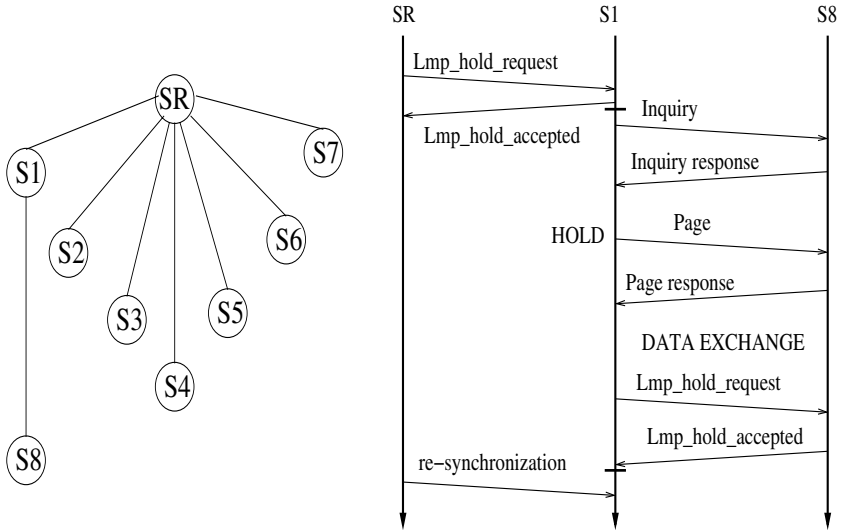


Fig. 2. Communication pattern for the first tree levels.

communication pattern for the SR and the first 2 levels of the tree. When SR has grafted the slaves S_1 through S_7 , it authorizes them to form new piconets. Then, SR puts its slaves into hold mode so that they can become active as masters. Before the *holdTO* imposed by SR to its slaves expires, S_1 must put its own slaves into hold mode. This way S_1 can return active in the SR piconet without its inactivity being interpreted as a disconnection by S_8 . While a piconet is under construction, the slaves it involves simply alternate between the active and the hold state.

3.2 Device Synchronization

To guarantee the tree connectivity we must carefully synchronize the BDs; the synchronization must be compliant with the Bluetooth specification. In figure 3, we show the lifecycle of a bridge, a connected slave of its still incomplete piconet and a free BD² throughout the tree construction. Those lifecycles have

² That is, a BD that does not belong to any piconet.

to fit together: a master that is forming its own piconet must periodically return active in the upstream piconet as a slave, according to the *holdTO* imposed by its own master. By contrast, when the SR's slaves are active in their own

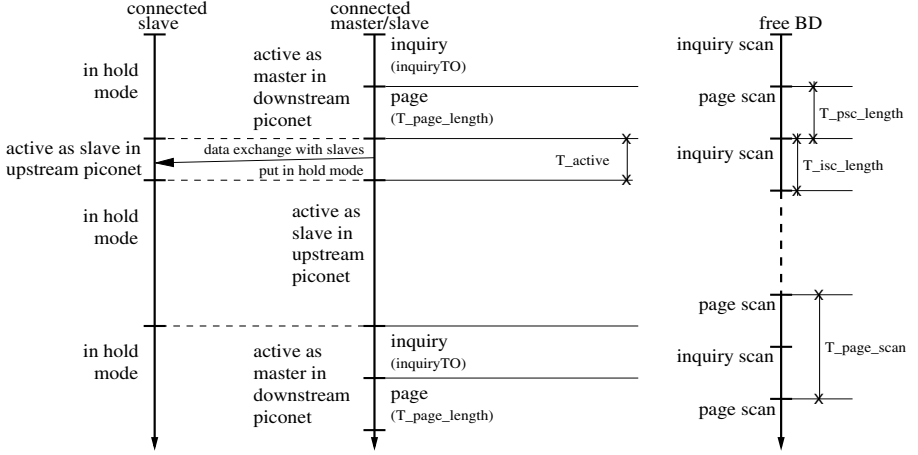


Fig. 3. Timers sizing.

piconets, SR remains inactive (fig.2).

Let us define the following time intervals:

- $inquiryTO$ is the interval in which a bridge performs the **inquiry** procedure;
- T_{isc_length} is the interval during which a free BD performs the **inquiry_scan** procedure. Since a free BD that receives an **inquiry** waits for a random time T_R ($0 < T_R < 1023$ slots, that is, $T_R \leq 0.64$ sec) before replying, it must be $T_{isc_length} > T_R$;
- T_{psc_length} is the interval during which a free BD either performs the **page_scan** procedure, if it has replied to an **inquiry** message in a previous **inquiry_scan** phase, or it stays inactive waiting to start a new **inquiry_scan** phase;
- T_{page_scan} is the interval between the beginnings of two consecutive **page_scan** phases for a free BD. It equals $T_{isc_length} + T_{psc_length}$;
- T_{page_length} is the interval during which a bridge performs the **page** procedure;
- T_{active} is either the interval in which a bridge is active as a master in its own piconet or the interval in which a BD is active as a slave in the upstream piconet.

The $inquiryTO$ and T_{page_scan} timers are defined as in the specifications. We can derive the above timer values from the Bluetooth specification, as follows. The minimum value for $inquiryTO$ is 1.28 sec., which allows to perform half of a train test. We adopt $inquiryTO = 1.28$ sec. To maximize the probability that a master and a free BD meet, thus minimizing the scatternet formation

latency, we choose the time spent in the **inquiry_scan** state (T_{isc_length}) equal to $inquiryTO$, that is, 1.28 sec. Moreover, we set $T_{psc_length} = 1.28$ sec.; this way, $T_{page_scan} = 2.56$ sec. If the interval between two consecutive **page_scan** phases is between 1.28 sec. and 2.56 sec., then the master shall use mode R2 to perform the **page** procedure, that is, it must repeat a train scan at least 256 times, thus spending 2.56 sec. which we adopt as the value of T_{page_length} . T_{active} must be long enough to allow the master to turn into hold mode its slaves, and to forward them some data. We choose to have the BDs active for 28 slots in each piconet: 14 slots are for the data exchange and 14 slots are for the hold mode negotiation. Hence, $T_{active} = (28 \cdot 0.625 msec.) = 17.5$ msec. A bridge stays active 17.5 msec. in the upstream piconet and 17.5 msec. in the downstream piconet. We also need a mechanism to de-synchronize the masters and the free BDs: if a master constantly performs the **inquiry** phase while some of the free BDs surrounding it are in the **page_scan** phase, they never connect. We introduce some randomization by having each BD that waits for a random time after the bootstrap before starting the first **inquiry_scan** procedure.

When a piconet is completed, the execution of the **inquiry** and **page** procedures is useless, and the activity time of a master can be completely exploited for the data exchange. In fig.4, we show the timers settings after the algorithm

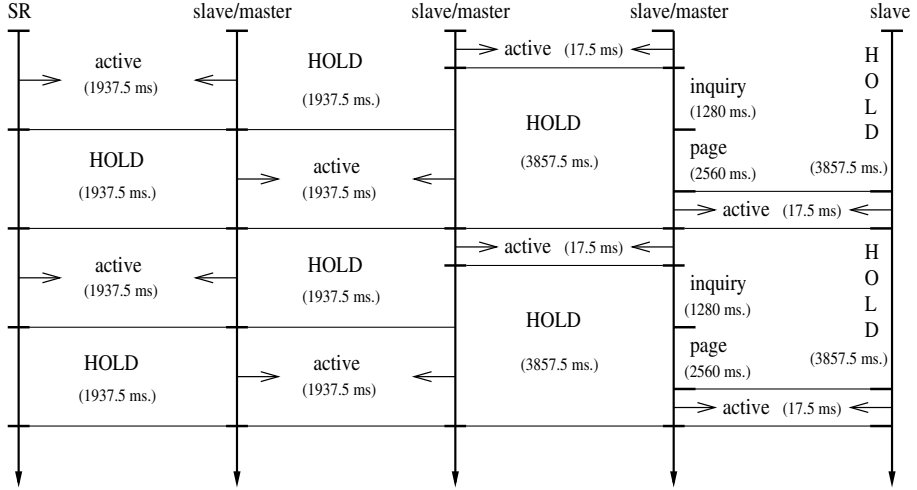


Fig. 4. Timers sizing after the scatternet has been built.

termination in the case of a 5-layers tree. The masters of the leaf piconets continue performing the **inquiry** and **page** procedures to support joining BDs, as we discuss in the next section. This requires an appropriate sizing of $holdTO$ at the above layer, that is achieved through the hold mode negotiation. The **inquiry** and **page** phases in the leaf piconets could be avoided only under the assumption that the number of BDs that must be connected to the scatternet

is a priori known, or that an agreement is maintained about the membership cardinality in spite of dynamic join and leave events. With this assumption, the whole bandwidth could be used for the data exchange once it is known that all the BDs have been grafted. Yet, we do not consider this assumption because in real environments the membership knowledge is in general not available.

3.3 Dynamic Membership and Topology

In this section, we describe the mechanisms to support dynamic changes in the scatternet topology and membership.

A bridge that leaves the scatternet partitions the ad hoc infrastructure leaving disconnected all the downstream piconets. The same occurs if a bridge moves out of the range of its upstream or downstream neighbors. To reconnect the scatternet, the disconnected subtree is *flushed*: a bridge noticing the disconnection from its parent³ notifies the failure to its own children and so on recursively, so that all the BDs in the disconnected subtree become free and they can rejoin. A free BD can graft the scatternet by becoming a slave in a not yet completed piconet.

It is more difficult to release the assumption concerning out-of-range BDs. In [8], the definition of *geographically connected* system is provided. A set of BDs is geographically connected if a (multi-hop) path exists between every two of them. This property is necessary to build a connected scatternet; anyway, it is not sufficient. If a master M exists having N free BDs in range, with N greater than the number of BDs M can connect, then M must select a subset of those BDs as slaves. The other free BDs could be out of the range of every other master, thus failing in connecting the scatternet.

As we will discuss in sec.4, the ability of exploiting the system geographical connectivity varies in the different algorithms. ODBT may suffer of the above syndrome. Anyway, the scatternet formation procedure may be modified to deal with out-of-range BDs while doing a best effort to form a connected infrastructure. According to the Bluetooth specification, in an error free environment the *inquiry* procedure must span at least 3 train switches (i.e., 10.24 sec.) to collect all the responses. We modify ODBT so that each master performs the *inquiry* procedure 8 times, before allowing its slaves to become bridges and to take forward the scatternet formation. This way, we still try to maximize the number of slaves per piconet, while at the same time preventing the algorithm from starving on a scatternet branch because of the lack of enough BDs in the range of a master. To dynamically graft moving and joining BDs, moreover, each master having less than 7 slaves periodically performs the *inquiry* and *page* procedures.

4 Performance Evaluation

In Table 1, we summarize the characteristics of ODBT and the other scatternet formation algorithms described in 2.1. The characterization of a tree topology,

³ Because of the *supervisionTO* expiration.

as done by ODBT, guarantees that no loops exist. The tree can be exploited as is for broadcast communications; otherwise, routing is easy to perform by appropriately forwarding the data along a subset of the tree branches. By contrast, when redundant paths exist between two BDs, a routing service is needed to avoid bandwidth waste in the message forwarding, but, the scatternet has a greater resilience to node mobility and failures. ODBT exploits M/S bridges only: according to [4], they allow to achieve lower inter-piconet delay with respect to S/S bridges. Differently from ODBT, the most part of the solutions proposed so far does not support BDs mobility or nodes leaving the scatternet, nor BDs joining after the procedure completes. Joining nodes can be supported provided that the masters of the incomplete piconets (i.e. those with < 7 slaves) periodically perform inquiries in order to discover them. The greater the number of incomplete piconets, the greater the probability that a free BD moves in a piconet that can graft it. On the other hand, the more populated are the piconets, the lower is the number of piconets, and the shorter are the paths within the scatternet. A short path requires a low number of bridge's switchings between adjacent piconets, thus providing a low latency. Bluetree and Bluenet explicitly bound the maximum number of slaves per piconet, but they are not able to exploit this characteristic to support dynamically joining devices. The goal of minimizing

Table 1. Comparison among scatternet formation algorithms.

	ODBT	Bluetree	Bluenet	BTCP	LMS	TSF
topology	tree	tree	graph	fully connected	graph	tree
bridge type	M/S	M/S, S/S	M/S, S/S	M/S, S/S	M/S, S/S	M/S
mobility support	yes	no	no	no	with extensions	yes
Join/Leave support	yes	no	no	no	with extensions	yes
dense piconet	≤ 7	5	N_{max}	< 7	6	\perp
# phases	1	2	3	3	1	1
out-of-range BDs	yes	yes	yes	no	with extensions	no

latency also motivates the choice of all the algorithms of having ≤ 7 slaves for each piconet: the management of the non-active slaves would prohibitively increase the re-synchronization complexity. On the other hand, apart from ODBT, all the considered works do not discuss in detail the problems involved by the BDs timer configuration for the switching between adjacent piconets. A performance index of a scatternet formation algorithm is the time spent to terminate before starting the data forwarding. Some algorithms require multiple phases, involving the discovery of the existing BDs and possible reconfigurations of the infrastructure characterized in the previous phases. Those reconfigurations may hinder the fast scatternet formation, thus negatively affecting the algorithm performance particularly in the case of mobility. By contrast, ODBT, LMS and TSF build the scatternet in one phase. The algorithms differ in the ability of dealing with out-of-range BDs. Those BDs cannot be supported by both BTCP and TSF. In the former algorithm, all BDs have to be in range to carry out the

coordinator election. In the latter algorithm, two neighbor components could be prevented from merging because the respective roots are out of range. Bluenet has problems similar to TSF, but its ability of merging two piconets exploiting any BD makes it more tolerant to out-of-range devices. The ODBT behaviour should be similar to that of Bluenet with respect to supporting out-of-range BDs. Bluetree seems to be the more effective solution to support out-of-range BDs, thanks to the capability of reorganizing the piconets. Notice that dealing with out-of-range BDs allows to build scatternets spanning a larger area, thus overcoming the limitations due to the short communication range provided by the Bluetooth technology.

4.1 Simulation Results

In this section we provide a quantitative analysis of ODBT by means of simulation techniques. ODBT has been implemented in the framework of the NS-2 network simulator, extended with Blueware [1] that simulates the Bluetooth protocol stack. Blueware includes the TSF implementation; this allows us to compare the behaviors of ODBT and TSF. So far, the simulation environment does not allow to reproduce BDs movements, nor it involves the notion of distance amongst BDs. As a consequence, we performed experiments only with in-range⁴ BDs that do not move.

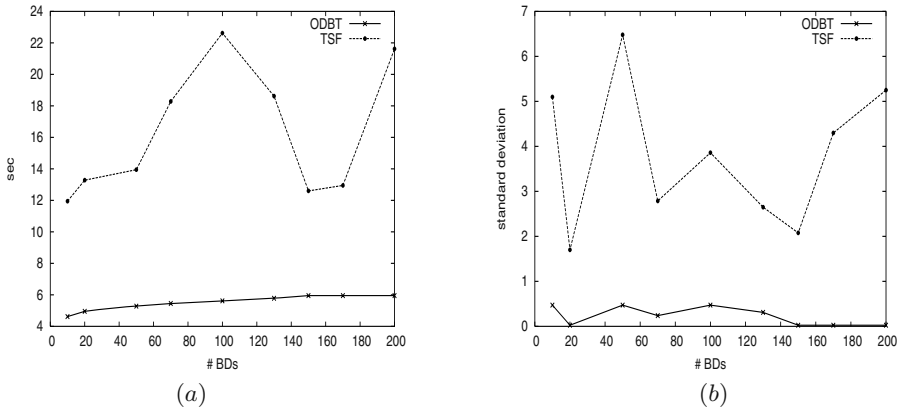


Fig. 5. (a) Termination time of the scatternet formation algorithms. (b) Standard deviation of the termination time.

The number of BDs varies between 10 and 200; the device with identifier 0 assumes the role of SR. We performed experiments with both BDs starting all at the same time and BDs starting sequentially every 2 sec., obtaining comparable results. The results reported in this section refer to the latter case; every experiment has been repeated 50 times and we report the average results.

⁴ This is a required TSF condition.

In fig.5(a), we show the average termination time. In TSF, when two components (either piconets or scatternets) meet, they can be merged into one. In our experiments, we observed that TSF tends to form pairs of BDs (a master connected to only one slave), that are then combined. This requires checking that the newly created links do not form loops, and possibly performing a master/slave switch to merge the two components. These operations are time consuming and they negatively affect the algorithm performance. Moreover, the algorithm behaviour is highly unstable: in fig.5(b), we show the standard deviation of the termination time. That behaviour depends on how the components form and

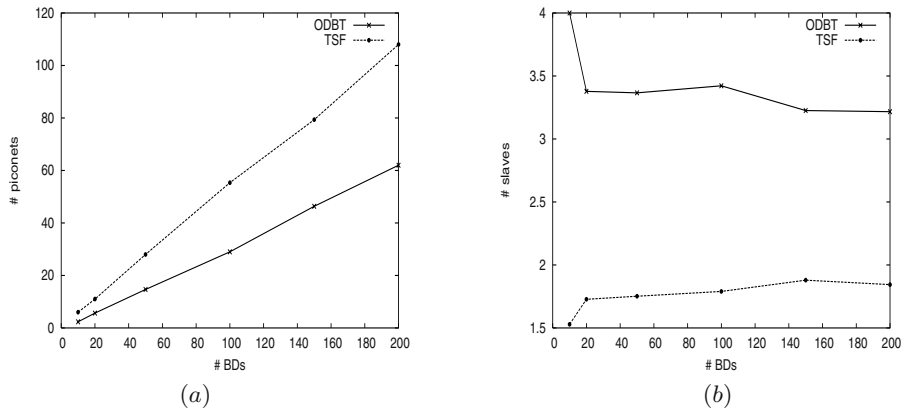


Fig. 6. (a) Average number of piconets forming the scatternet. (b) Average number of slaves per piconet.

merge throughout the simulation, depending on random events that make the algorithm performance not predictable. This characteristic may be a flaw with respect to the service provided to the users, in that it delays the data forwarding particularly to the last connected BDs, that can observe data loss if the transmission starts while the scatternet formation is still ongoing.

ODBT builds more dense trees than TSF: in fig.6(a), we show the average number of piconets composing the scatternet, while in fig.6(b) we report the average number of slaves per piconet. In the ODBT case, indeed, all the high level piconets have 7 slaves in the adopted experimental setting; the average is decreased by the leaf piconets. Having the slaves in a piconet that contemporarily start behaving as bridges allows to obtain balanced trees: in our simulations we observed a maximum difference of one level among the leaf piconets. By contrast, the random component merging in TSF tends to form low populated piconets that concatenate to each other yielding deep trees (fig. 7(a)).

Minimizing both the tree depth and the number of piconets allows to minimize the number of bridges as well. Bridges represent the bottleneck in the data forwarding, as a bridge receiving data from its upstream master has to

wait that the master puts it into hold mode, before acting as a master on its own and forwarding the data in its downstream piconet. We evaluated the la-

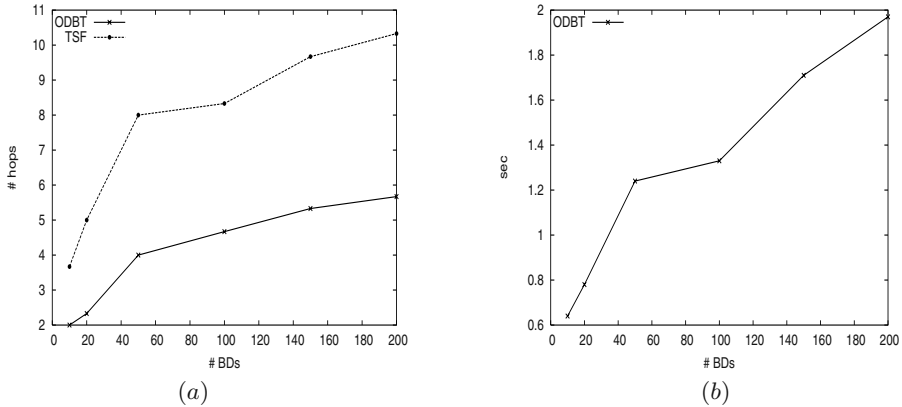


Fig. 7. (a) Average tree depth. (b) Maximum latency in the data forwarding with ODBT.

tency perceived at the farthest BD from the SR, after the scatternet has been formed, averaged on 1000 data packets (fig.7(b)); the reported results refer to the complete infrastructure where the leaf masters do not execute the **inquiry** and **page** procedures. The latency increases with the number of involved BDs proportionally to the increase of the tree depth.

5 Concluding Remarks

In this work we present the On-Demand Bluetooth scatternet formation protocol (ODBT) for the formation of scatternets with a tree topology. We show how the Bluetooth stack should be configured to implement ODBT. We compare ODBT with other algorithms existing in the literature, also by means of simulation techniques.

ODBT builds efficient tree structures, minimizing the number of bridges connecting the piconets and relying only on M/S bridges. ODBT shows a stable behaviour for increasing number of involved BDs. It is able to dynamically re-configure the scatternet to deal with joining, leaving and moving BDs, and it can support out-of-range devices.

We are currently extending the simulator with the notion of distance amongst BDs and the simulation of movements, to perform further experiments with ODBT in order to evaluate its behavior in the presence of out-of-range moving devices. In particular, we are interested in observing how the data transmission latency varies as a function of the number of slaves that are grafted in the piconets.

As a future work, we plan to extend the protocol so that it can operate in the presence of multiple SRs. In the case different devices contemporarily start the formation of a scatternet involving the same BDs, we must be able to merge the different scatternets into one to guarantee the system connectivity. This can be achieved by including a SR election algorithm in ODBT. Further developments concern the implementation of unicast and multicast routing services on top of ODBT, and the deployment of ODBT on a testbed platform.

References

1. Balakrishnan H., Gutttag J., Miu A., Tan G.: “*Blueware: Support for Self-organizing Scatternets*”. <http://nms.lcs.mit.edu/projects/Blueware/>.
2. Bluetooth Special Interest Group: “*Bluetooth V1.1 Core Specifications*”. (May 2001). <http://www.bluetooth.org/>.
3. Law C., Mehta A.K., Siu K.-Y.: “*A New Bluetooth Scatternet Formation Protocol*”. Proc. IEEE Global Telecommunications Conference (GLOBECOM’01), 2001, 2864–2869. <http://perth.mit.edu/ching/pubs/ScatternetProtocol.pdf>.
4. Misić V.B., Misić J.: “*Performance of Bluetooth Bridges in Scatternets With Exhaustive Service Scheduling*”. Proc. 36th Hawaii International Conference on System Sciences (HICSS’03).
5. Salonidis T., Bhagwat P., Tassioulas L., LaMaire R.: “*Distributed topology construction of Bluetooth personal area networks*”. Proc. IEEE INFOCOM 2001, Vol. 3 (2001) 1577–1586.
6. Tan G., Miu G., Gutttag J., Balakrishnan H.: “*An Efficient Scatternet Formation Algorithm for Dynamic Environments*”. IASTED International Conference on Communications and Computer Networks (Nov. 2002)
7. Wang Z., Thomas R.J., Haas Z.: “*Bluenet - a new scatternet formation scheme*”. Proceedings of the 35th Annual Hawaii International Conference on System Sciences (Jan. 2002). <http://wnl.ece.cornell.edu/Publications/hicss02.ps>.
8. Zaruba G.V., Basagni S., Chlamtac I.: “*Bluetrees-scatternet formation to enable Bluetooth-based ad hoc networks*”. Proc. IEEE International Conference on Communications (ICC 2001), 2001, 273–277.

GPS-Based Route Discovery Algorithms for On-Demand Routing Protocols in MANETs

Mehran Abolhasan and Tadeusz Wysocki

Telecommunication and Information Research Institute,
University of Wollongong, NSW 2522, Australia
`mehran@titr.uow.edu.au`, `wysocki@uow.edu.au`

Abstract. This paper presents new Global Positioning System (GPS)-based route discovery algorithms for on-demand routing in MANETs, called Position-based Selective Flooding (PSF). We applied our route discovery algorithm to our previous routing protocol, which is called Location-based Point-to-point Adaptive routing (LPAR) protocol and investigated its performance by simulation. Simulation results show that our position based flooding algorithm produces fewer routing overheads than the pure flooding, expanding ring search (used in AODV), LAR1 and our existing LPAR strategy, as network traffic and density is increased. Furthermore, we propose a number of improvements and variations which can be used instead of, or to further improve the performance of PSF under different network conditions.

1 Introduction

Mobile Ad Hoc Networks (MANETs) have become one of the most highly researched areas in wireless local area networking. The current research in MANETs involves all layers in the TCP/IP model, as the very nature of these networks demands new design rules for each layer in order to provide efficient end-to-end communications. MANETs are made up of a number of nodes, such as laptops and Personal Digital Assistance (PDA), which are capable of communicating with each other without using a fixed base station. This means that each node performs routing in a distributed manner. The limitation in transmission range and the highly dynamic nature of these networks, makes data transmission between the source and the destination travel over multiple hops, which can vary over time. Therefore, routing in MANETs is a challenging task. In the MANET literature, a number of different routing protocols have been proposed. Designing routing strategies for MANETs began by optimising the routing protocols designed for wired networks[7]. The route discovery in these protocols were proactive in nature, which means that each protocol periodically exchanged routing information with other nodes in the network, in order to build their routing tables. However, this approach to routing lacks scalability as the size of the network grows. On-demand routing protocols were designed to reduce the route discovery overheads by allowing each node to determine routes when

they are required, rather than maintaining a route to every destination. This routing strategy consists of two phases: route discovery and route maintenance.

A node which requires a route to a particular destination, starts a route discovery phase, where a Route Request (RREQ) packet is propagated through the network until the destination or an intermediate node to the destination is found or the packet expires. When a route is found, a Route Reply (RREP) is sent back to the destination using link reversal if the RREQ has travelled over bidirectional links or by flooding if unidirectional links are used. The route maintenance phase is initiated by an intermediate node, which experiences a link failure while a route is still active. In this phase, the route can be repaired locally at the point of failure by using a localised route maintenance strategy [4] or the node which detected the link failure notifies the source via a Route Error (RERR) message and the source will either use another route or initiate another route discovery [6].

Routing in on-demand protocols can be classified into two groups: source routing and point-to-point (also called hop-by-hop) routing. In source routing protocols such as [6], each data packet carries the complete source to destination address, whereas in point-to-point routing [4][3], data packets only carry the destination address and the address of the next hop which leads to the destination. This means that each intermediate node in an active route can make routing decisions, thereby allowing active routes to be adaptable to topology changes, whereas in source routing all the routing decisions are made at the source. This means that link failure in an active route may result in initiation of additional route discoveries at the source or at the point of failure¹. Furthermore, in source routing, an increase in the number of hops in the active route will result in an increase in the amount of overhead carried by each packet. In contrary, in point-to-point routing the size of each packet is not affected by multihopping. Therefore, point-to-point routing has more potential to scale better as the size of the network increases.

In our previous study of point-to-point routing [3], we proposed Location-based Point-to-point Adaptive Routing (LPAR). In this study, we introduce a new approach to reduce route discovery overheads, given that each source node possesses location information about the required destination. Furthermore, in LPAR, we proposed a number of different strategies to minimise the effects of link failure on the active route and increase the stability of each route. In this paper, we introduce new strategies to reduce route discovery overhead, while maintaining high levels of throughput when the source has no location information about the destination. We implemented our route discovery strategy on top of LPAR and compared its performance with AODV and LAR1 using simulation. Our results show that this new approach has fewer overheads than AODV and LAR1, and has higher levels of scalability as the size (i.e. boundary), node density and traffic in the network grows. The rest of this paper is organised as follows. Section 2, describes our route discovery strategy. Section 3, describes the simulation environment and the parameters used. Section 4, presents a dis-

¹ If a localised route repair strategy is used.

cussion on our simulation results. Section 5, presents a number of alternative strategies and improvements for our routing strategy and section 6 presents the concluding remarks.

2 Proposed Strategy

In this section, we propose Position-based Selective Flooding (PSF). In pure flooding or in ERS, all the neighbouring nodes usually rebroadcast the RREQ message, unless the TTL has expired. In a dense network, routing overhead can be significantly reduced by strategically selecting the retransmitting nodes to cover the entire network (or a selected area). In PSF, only a number of different nodes forward the RREQ packet, based on a selection criteria described below. We have also proposed a number of variations and improvements to make PSF more efficient.

2.1 Overview and Definition

This strategy reduces the number of re-broadcasts during route discovery by allowing nodes, which are positioned in a determined region, to re-broadcast the routing packets. To illustrate how this strategy works, suppose node S (see Figure 1), wants to determine a route to node D. Node S will initiate its route discovery, and a RREQ is broadcasted, which stores the source nodes location information. The receiving nodes then determine their relative distance to node S and rebroadcast the RREQ if they map into the Forwarding Region (FR). Note that the idea behind choosing FR comes as a result of the following observations:

1. Nodes that are located near the boundary of the transmission range, R , will create unstable (or short lived) links if they are selected as intermediate nodes in an active route.
2. Selection of intermediate nodes which are close together will increase the number of hops in each route. This means that end-to-end delay will increase during data transmission. Furthermore, probability of route failure may increase. Since the number of intermediate nodes in an active route increases, then the probability of a link failure causing the route failure will increase.
3. In a dense network, flooding over neighbours which are very close to each other may not significantly increase the probability of a successful route discovery or searching the entire network. In this case, routing overhead can be significantly reduced by strategically selecting the rebroadcasting nodes.

Each node, which receives a rebroadcasted RREQ packet, will also calculate their own FR. If their location coordinates map within the FR and they are further away from the source than the previous hop, they will rebroadcast the RREQ packet. This is done by (see Figure 1) multiplying R_{max} by the hop count and setting $R_{min} = R_{max} - K$. Therefore, the RREQ packet will continue to propagate away from the source at each hop. Note that K is a variable, which

determines the width of FR. In our simulation, we used a constant value for K . However, in section 6, we propose a number of different strategies, which can be used to dynamically select values for R_{max} and K at each hop by taking into account the location and the number of neighbouring nodes, for each node. The advantage of PSF is that RREQ packets do not need to carry a forwarding list² to limit the number of rebroadcasts, as compared to the neighbour aware strategies such as MPR. This means that the size of each RREQ packet will be smaller. Furthermore, nodes do not need to maintain 2-hop topology information.

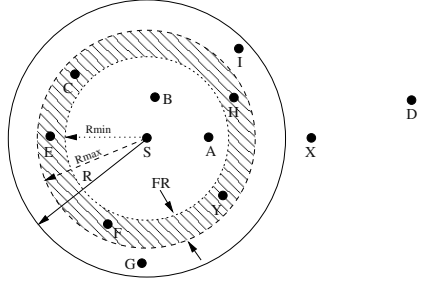


Fig. 1. Illustration of the forwarding region

2.2 Mathematical Model

In PSF, a node rebroadcasts a RREQ message if it satisfies the following three conditions:

1. The rebroadcasting node must be further away from the source than the downlink node it received the RREQ from.
2. The RREQ packet has not been seen before, or it has not expired
3. The forwarding node must lie within the FR.

We calculate FR at each node, such that condition one is also met. To show how this is achieved, assume that the MANET topology can be seen as an undirected graph $G = (V, E)$, where V represents a set of mobile nodes connected by a set of edges (i.e. links), E , if the distance between two nodes is less than R . Each node i , in G , has a set of neighbours n_x such that³ $d(i, n_x) \leq R$. Let h be the hop count. At each hop the nodes must satisfy the following condition in order to be able to rebroadcast.

$$(r_{min} = r_{max} - K) \leq d(i, n_x) \leq (r_{max} = C * h) \quad (1)$$

For $h = 1 \dots N$

² A list of rebroadcasting nodes

³ Assume all nodes have equal transmission range

2.3 Theoretical Overhead Analysis

In PSF, the number of rebroadcasting nodes are lower than in pure flooding at each hop, when the size of the forwarding region is less than the maximum transmission radius. To show this, let $N_{max} = |N_x|$ be the maximum number of neighbours for a particular node, and let N_{tx} be the number of retransmitting nodes. Now assume that all nodes are equally distributed in the network and all nodes have equal transmission range. In pure flooding the flooding area for a particular node is πR^2 and the number of retransmitting nodes is equal to N_{max} (i.e. $N_{tx} = N_{max}$). In PSF, the flooding area (or the forwarding region) is $A_{FR} = \pi r_{max}^2 - \pi r_{min}^2$. Let $L_{FR} = r_{max} - r_{min}$ be the width of the FR. Then, it can be easily seen that, in PSF, $N_{tx} \leq N_{max}$ for $L_{FR} \leq R$, or:

$$\lim_{L_{FR} \rightarrow R} A_{FR} = \pi R^2 \quad (2)$$

Therefore, only in the worst-case scenario, where $L_{FR} = R$, PSF will converge to pure flooding, and for cases where the required destination could be easily found with $L_{FR} \ll R$, then $N_{tx} \ll N_{max}$. This means that the number of RREQ packets propagating through the network will be far lower. To illustrate this with an example, suppose node S (see figure 2), wants to find a route to node D, and assume node S initiated a route discovery with a Time To Live (TTL) of 2, indicating that the RREQ packet can only travel over 2 hops. Suppose that the FR is calculated as shown in figure 2, which shows that nodes H, Y and M are in FR. Therefore, only these nodes rebroadcast the RREQ. When node I, receives a RREQ from node H, it will send a RREP back to the source, using link reversal⁴. This will then bring the total number of broadcast to 6 (i.e. 4 RREQs and 2 RREPs). Now suppose all nodes within R where able to retransmit, then the total number of broadcasts will be 12 (i.e. 10RREQs and 2RREPs). Therefore, in this simple scenario, PSF produced 50% less control packets than pure flooding.

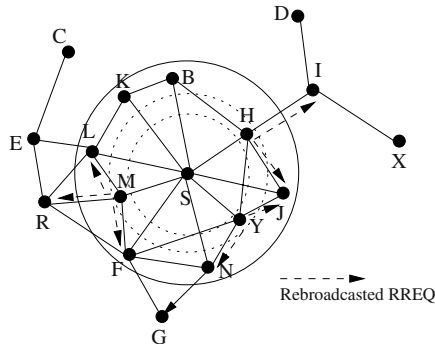


Fig. 2. An example of route discovery using PSF

⁴ assuming only bidirectional links are used

3 Simulation Environment and Performance Metric

We performed our simulations using the GloMoSim[1] simulation package. GloMoSim is an event driven simulation tool designed to carry out large simulations for mobile ad hoc networks. Our simulations were carried out for 100, 200, 300, 400 and 500 node networks, migrating in a 1000m x 1000m area. IEEE 802.11 DSSS (Direct Sequence Spread Spectrum) was used with maximum transmission power of 15dbm at a 2Mb/s data rate. In the MAC layer, IEEE 802.11 was used in DCF mode. The radio capture effects were also taken into account. Two-ray path loss characteristics was considered as the propagation model. The antenna height was set to 1.5m, the radio receiver threshold was set to -81 dbm and the receiver sensitivity was set to -91 dbm according to the Lucent wavelan card[2]. Random way-point mobility model was used with the node mobility ranging from 0 to 20m/s and pause time varied from 0 to 900s. The simulation was run for 900s for 10 different values of pause time, and each simulation was averaged over eight different simulation runs using different seed values. Constant Bit Rate (CBR) traffic was used to establish communication between nodes. Each CBR packet was contained 512 Bytes and each packet were at 0.25s intervals. The simulation was run for 10 and 20 different client/server pairs⁵ and each session was set to last for the duration of the simulation. In our simulation study of PSF, we set constants, which are used to calculate R_{max} and R_{min} (i.e. C and K), to 300m and 150m respectively. Therefore, the length of FR, $L_{FR} = 150m$. This simple model was used in our simulations to show the benefits of PSF in medium to large networks. In section 6, we propose a number of strategies to dynamically select values for C and K to increase the efficiency of the algorithm under different levels of node density.

The performance of each routing protocol is compared using the following performance metrics.

- Packet Delivery Ratio (PDR)
- Control (O/H)
- End-to-End Delay

PDR is the Ratio of the number of packet sent by the source node to the number of packets received by the destination node. Control (O/H) presents the number of routing packets transmitted through the network for the duration of the simulation. This metric will illustrate the levels of the introduced routing overhead in the network. Finally, the End-to-End Delay metric illustrates the average end to end delay for transmitting one data packet from the source to the destination.

4 Results

In this section, we present a discussion on our simulation results. Note that we implemented the PSF strategy on the top of our existing routing protocol,

⁵ Note that the terms Client/Server, src/dest and Flows are used interchangeably

which is called LPAR, and we refer to this as LPAR-PSF. The performance of LPAR-PSF was compared with LPAR-S⁶, LPAR, LAR1 and AODV.

4.1 Packet Delivery Ratio Results

Figure 3 and 4, show the PDR for a 100 node and 500 node network, with 10 src/dest pairs. These results, illustrate the performance of the protocols in a moderately dense and a highly dense network. In the 100 node scenario all protocols achieve over 95% PDR during the high mobility phase, where the pause time is low, and achieve over 97% PDR for mid-range mobility to zero mobility. However, in the 500 node scenario, the point to point based routing strategies outperform the source routing strategy. This is more evident under high levels of mobility, where LAR1, under-performs the other strategies. This can be due to a number of different reasons. Firstly, when a route failure occurs, in LAR1, a source node scans its route cache to use an alternate route. Under low levels of mobility, the routes in the route cache will stay active longer than under high levels of mobility. This is because the probability of a link failure in an active route (since complete source to destination address is used) increases in high mobility, which means that more route discoveries will be initiated at the source and more packets may be dropped in the process. This may become more evident as the amount of multihopping increases within each route. Furthermore, in the point to point routing strategies (i.e. LPAR and AODV) each intermediate node in an active route can learn and use a better route to the destination, which means that the route between the source to the destination may stay valid for longer time, whereas in LAR1, since each data packet relies on the source to destination address given in the header, a link failure in an active route may immediately cause a route failure, which require a RERR message to be sent back to the source where another route must be calculated or used (if available).

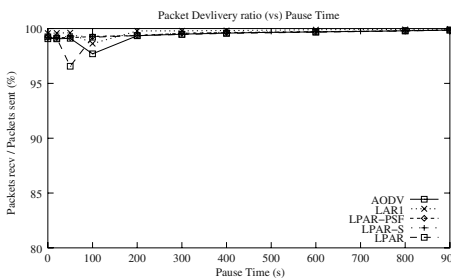


Fig. 3. PDR: 100N, 10 Flows

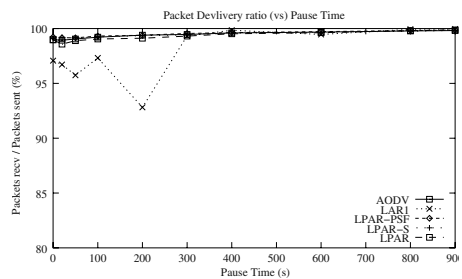


Fig. 4. PDR: 500N, 10 Flows

Figure 5 and 6, show the PDR for a 100 node and 500 node network, with 20 src/dest pairs. In the 100 node network, all protocols produce over 90% PDR.

⁶ This is the LPAR routing strategy, which also selects stable routes, in our previous study we referred to this as LPAR-S[3]

The point-to-point routing protocols perform very consistently over the different levels of mobility, in particular LPAR, LPAR-S and LPAR-PSF maintain over 97% PDR for all levels of mobility. LAR1, slightly under-performs under high mobility, where its performance drops to 90% for the 20 second pause sample. In the 500 node network, the performance of each routing strategy can be clearly distinguished. Here, LPAR-PSF has the best performance, where it maintains over 98% PDR. LPAR and LPAR-S also produce over 95% PDR. However, AODVs performance significantly drops under high mobility when compared to the 100 node scenario. This clearly highlights the advantage of exploiting location information to generate a more strict route discovery procedure and reduce bandwidth consumption in highly dense networks. LAR1 again shows the worst performance under high mobility, where its PDR varies between 80 to 85%.

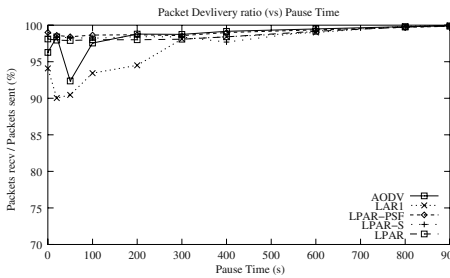


Fig. 5. PDR: 100N, 20 Flows

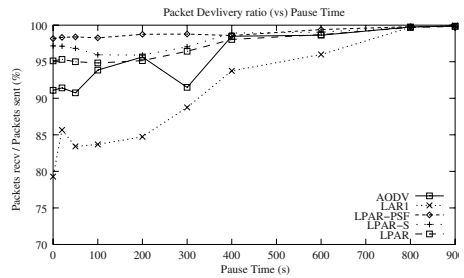


Fig. 6. PDR: 500N, 20 Flows

4.2 Control Overhead Results

Figure 7 and 8 show the number of control packets introduced into the network by each routing protocol, for 10 CBR sources, in a 100 node and a 500 node network respectively. AODV produces more control packets than all other routing strategies. This is more evident under high mobility, where AODV produces up to 10000 more control packets than its nearest competitor (i.e. LAR1). Two factors contribute to reducing routing overhead in LAR1 when compared to AODV. Firstly, nodes can have multiple routes to destinations stored in a route cache (as discussed earlier), which may reduce the number of route discoveries initiated for each src/dest pair, whereas in AODV, each node only stores a single route. Secondly, in LAR1, if source nodes have location information about the required destination, they can use RZS (as described earlier), which minimises (or localises) the search area to a particular region. The advantage of this is that the number of nodes involved in broadcasting RREQ packets is reduced, which means that fewer control packets are transmitted. This also allows more bandwidth to be available for the nodes that are not in the search area and reduce channel contention. LPAR and LPAR-S, which use the 3-state

route discovery algorithm, produce less overhead than LAR1, despite only storing single routes. This is because in our 3-state route discovery algorithm, if unexpired location information is available, the source node will first attempt to discover a route by unicasting rather in broadcasting (as previously described in section 2.4). Hence, fewer control packets are transmitted through the network. LPAR-S further reduces this overhead by flooding over links which have certain level of stability. The advantage of this is that the route may last longer, which means fewer route recalculations will be required and fewer data packets will be dropped. LPAR-PSF produces fewer control packets than all other routing strategies. In the 100 node network LPAR-PSF produces upto 2500 less control packets than LPAR-S and up to 17000 packets less than AODV in high mobility. Under higher node density (i.e. the 500 node scenario), LPAR-PSF produces up to 20000 less packets than LPAR-S and up to 110000 less packets than AODV. In the 20 src/dest scenario (see Figure 9 and 10), the gap between LPAR-PSF and the other strategies increases, particularly during high levels of mobility. It can be seen here that LPAR-PSF, produces 40000 less control packets than LPAR-S and up to 500000 less control packets than AODV. This clearly, shows the benefits of using PSF in highly dense networks.

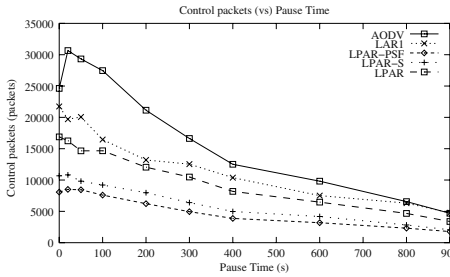


Fig. 7. CTRL: 100N, 10 Flows

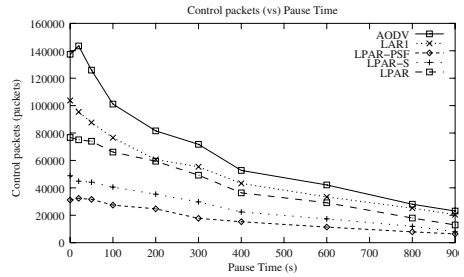


Fig. 8. CTRL: 500N, 10 Flows

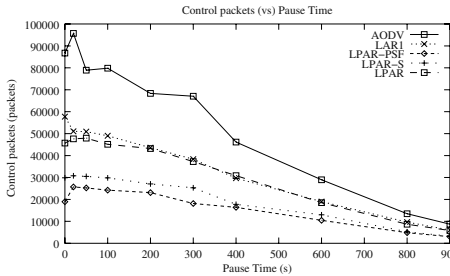


Fig. 9. CTRL: 100N, 20 Flows

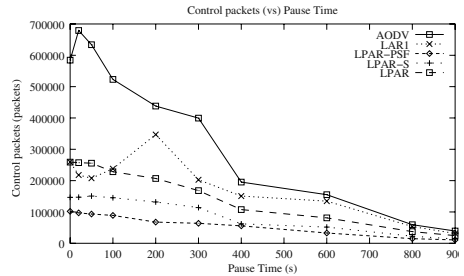


Fig. 10. CTRL: 500N, 20 Flows

4.3 Delay Results

Figures 11 and 12 show the average end-to-end delay experienced by each data packet for a 100 and a 200 node network with 10 src/dest pairs. As expected, all protocols experienced larger delays during high mobility, since more frequent link failure may cause route recalculation. This means that packets may experience longer delays before they reach their destination. In the 100 node network scenario, AODV has lowest end-to-end delay when compared to the other protocols. This is because, AODV always uses the shortest route to the destination and it only maintains a single route, whereas LAR1 can store multiple routes. This means that if the optimal route fails (the one with the shortest src/dest path), an alternate route from the route cache may be used. Therefore, some packet may travel over longer routes to reach the destination. Similarly in LPAR and LPAR-S and LPAR-PSF a secondary route may be available for each active route [3]. Therefore, if the primary route fails, some packets may travel over the secondary route, which may be longer in length. Hence, they may experience slightly longer delays. From the figure 11, we can see that LPAR and LPAR-S have on average about 5ms more delay across all ranges of mobility. However, by using a secondary route, LPAR and LPAR-S are able to successfully transmit more data packets, and reduce the number of route recalculations, which means fewer control packets. In the 200 node scenario, the gap between AODV and the other routing strategies becomes smaller. This is because with a higher density more nodes are contending for the medium. Therefore, since AODV produces significantly more overhead than the other strategies, it will introduce more channel contention and consume more bandwidth than the other strategies. Hence, longer delays may be experienced by intermediate nodes in active route before they can gain access to the medium. In LPAR-PSF, however, significantly fewer control packets are transmitted than in AODV. Furthermore, only a number of selected nodes rebroadcast, which means that there will be less channel contention than in AODV. Therefore, in LPAR-PSF, even though the packets may travel over more hops, they still experience similar levels of delay when compared to AODV.

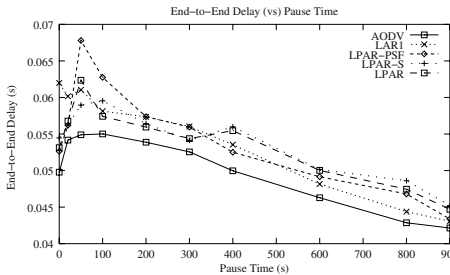


Fig. 11. Delay for 100 nodes

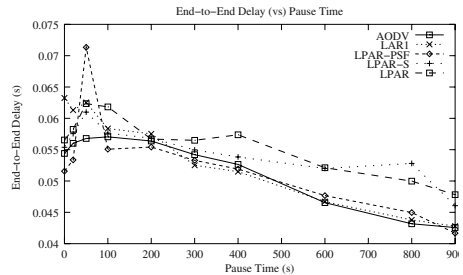


Fig. 12. Delay for 200 nodes

5 Alternative Strategies and Improvements

In our simulation study of PSF, we chose two constant values for C and K for every node to calculate the R_{max} and R_{min} in the PSF algorithm. This was done to illustrate the benefits which PSF could have in a medium to large network. However, selecting constant values may not be beneficial for every network size or topology. For example, chosen values for C and K for a large network may not produce good results in a small network and vice versa. Furthermore, we want every node to be able to calculate different FR's according to the node density of their neighbouring topology. Thus being able to successfully forward the RREQ message to different parts of the network, while minimising the number of re-transmitting nodes. In this section, we present a number of different strategies to dynamically select FR's at each node. Furthermore, we present an alternative strategy, which may also reduce route discovery overheads.

5.1 Source-Driven FR Selection

One way to introduce variable FR in the PSF algorithm is to determine its size at the source. We call this strategy Source-Driven FR selection or FR-SD. In this strategy, the source node specifies the values for C and K to be used at each hop. During the route discovery phase, the source node determines values for K and C and includes these values in the RREQ packet. Each time the route discovery fails (or the source node times out and no route is found), the distance between K and C, or R_{max} and R_{min} is increased and another route discovery is initiated. This process continues until a route is found or the distance between K and C becomes equal to R (i.e. $D_{KC} = R$). The *FR-SD* algorithm is outlined below.

Algorithm *FR-SD*

(* The FR-SD algorithm *)

1. $A \leftarrow$ Average Source/neighbour distance
2. $D_{AR} \leftarrow R - A$ (* distance between R and A *)
3. $P \leftarrow \{0.25, 0.5, 0.75, 1.0\}$ (* % inc for K and C *)
4. **for** $i \leftarrow 0, i \neq 5, i++$
5. $C \leftarrow A + D_{AR}P_i$
6. $K \leftarrow A - AP_i$
7. initiate Route Discovery
8. wait for reply
9. **if** *Route = found*
10. break loop
11. initiate data transmission

In the FR-SD algorithm, the value of A can be determined by employing any of the following methods⁷

1. Assume a maximum number of nodes for a network with known area, then A can be approximated for each node[5]

⁷ Methods 1 and 2 can assume an equal node distribution

2. If the source has location information about every neighbour, then $A = \frac{N_T}{D_T}$, where N_T is the total number of neighbours and D_T is the total distance.
3. Assuming each node calculates A, and exchanges it with its neighbours using hello beacon messages, from the collected values of A from every neighbour, each node can calculate an average value of A. This will give an average neighbour distance per node for a 2-hop region.

5.2 Distributed Node-Density Based FR

In the previous strategy (i.e. FR-SD), the size of C and K were determined and enforced by the source. The disadvantage of this is that the calculated values for C and K may not result in optimal retransmission and coverage at every forwarding node. In this strategy, we attempt to make the selection of C and K distributed (hence the name FR-DN), thus allowing each forwarding node to determine an optimal FR for itself. To describe how this strategy works, suppose that node S initiates a route discovery, and broadcasts a RREQ packet, which includes its location information and FR. The receiving neighbours check to see if they lie in the FR according to the information in the received RREQ packet. If this is true, they calculate their own FR, which replaces the existing FR in the received RREQ packet. Similarly, the nodes which receive the rebroadcasted RREQ packet check to see if they lie in the previous hop's FR. If they do and they have not seen the packet before, then they calculate their own FR and rebroadcast the RREQ as long as they are further away from the source than the previous hop. To determine C and K for R_{max} and R_{min} at each hop, we calculate A as before (in FR-SD), then calculate $D_{AR} = R - A$, $C = A + D_{AR}P$ and $K = A - AP$. However, this time we want P to be inversely proportional to the node density N_d , where P is varied from 0% to 100%. To do this, we assign a minimum FR density threshold (D_{FR}), where we increase the size of P until we get a minimum number of nodes falling within the FR. Note that we can set D_{FR} to be a certain percentage of N_d , so that for large values of N_d , a small FR will be selected, and for a small N_d , a large FR will be selected. We also want $L_{FR} \propto \frac{1}{N_d}$. One way to determine this is to use a hyperbolic function such as, $f(x) = \frac{c}{x+1}$ or a sigmoid function, such as $f(x) = \frac{1}{1+e^{ax}}$, to scale the required D_{FR} according to the node density. To illustrate how a sigmoid function can be used to calculate the percentage difference, P, between C and K, let N_{max} be the maximum possible number of neighbours at each node and P be equal to the sigmoid. Then we want our sigmoid function to vary between 0 and 1, such that for $N_d \rightarrow N_{max}$ our sigmoid, $f(N_d) \rightarrow 0$ and for $N_d \rightarrow 0$, $f(N_d) \rightarrow 1$. Therefore, we let $f(N_d) = \frac{1}{1+e^{0.05(N_d - \frac{N_{max}}{2})}}$. Figure 13 illustrates the sigmoid function for $N_{max} = 100$. From this figure, it can be seen that as N_d approaches 100 the sigmoid starts to approach 0, and as N_d approaches to 0 the sigmoid starts to approach 1. This characteristic will allow us to scale P according to the node density, which varies between 0 and 1 (or 0% to 100%).

Another possible method, would be to use the standard deviation of A (the average neighbour distance), to vary the FR.

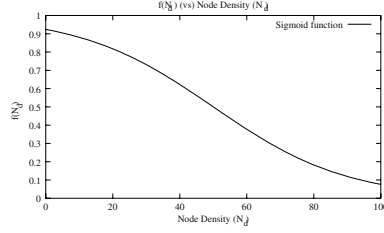


Fig. 13. Determining the percentage difference for FR using a sigmoid

5.3 Directional Node-Density

In FR-DN, the size of the FR is varied according to node density, by ensuring that a minimum number of nodes is present in the FR. However, FR-DN does not take into account the location of the nodes within FR. Hence, we introduce directional node density. The idea behind this strategy is to select the smallest L_{FR} that will contain a set of nodes which are located in a number of different parts of the FR. That is, we start with a small FR and increase its area until a number of strategically located nodes can be found within the FR.⁸ To do this, we select a number of sample point (this can be any number of points, e.g. 4 point to represent North, East, South and West), which are A meters away from the source (like before A is the average node to neighbour distance), as shown in Figure 14. We then check to see if there is a set of nodes within the current FR, which are closely located (by a threshold distance that can be optimised by using it as a simulation parameter) to each of these points. If there are at least one node close to every point, the current FR will be accepted as large enough to be able to rebroadcast the RREQ message away from the source and to different parts of the network. If not, the size of the FR is increased, and the new FR is scanned to see if the new boundary will host nodes, which are close to the required points.

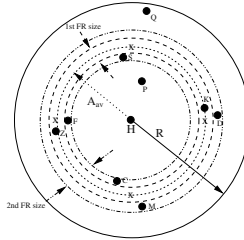


Fig. 14. Illustration of FR selection using Directional Density

⁸ Note that this strategy is an optimisation for FR-DN. Hence, it must be built on top of FR-DN.

To illustrate this, suppose node H (see Figure 14), calculates the 1st FR as shown. From the figure, it can be seen that within this FR only two nodes can be found near the required point (i.e. nodes K and Z). However, in the 2nd FR increase, at least one node is found near each required point (i.e nodes S, K, D, F, Z, C and M). Therefore, node H will use the second FR. The advantage of this strategy is that the FR ensures that the RREQ packet is propagated to different part of the network, and guarantees a certain level of coverage at each hop.

6 Conclusion

This paper presented a new routing discovery strategy for mobile ad hoc networks. We presented Position-based Selective Flooding (or PSF). In this strategy, only a number of selected nodes take part in route discovery. We implemented PSF on the top of LPAR and we referred to as LPAR-PSF. We compared its performance with LPAR, LPAR-S, LAR1 and AODV using simulations. Our results show that LPAR-PSF produces fewer overhead packets than, LPAR, LPAR-S, LAR1 and AODV, and achieves the highest levels of throughput in medium to large networks.

References

1. Glomosim scalable simulation environment for wireless and wired network systems. In <http://pcl.cs.ucla.edu/projects/glomosim/>.
2. Orinoco pc card. In <http://www.lucent.com/orinoco>.
3. Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. LPAR: An Adaptive Routing Strategy for MANETs. In *journal of Telecommunication and Information Technology*, pages 28–37, 2/2003.
4. S. Das, C. Perkins, and E. Royer. Ad Hoc On Demand Distance Vector (AODV) Routing. In *Internet Draft, draft-ietf-manet-aodv-11.txt*, work in progress, 2002.
5. Horst Hellbrück and Stefan Fischer. Towards Analysis and Simulation of Ad-Hoc Networks. In *ICWN02, pages 69–75, Las Vegas, Nevada, USA, June 2002*.
6. D. Johnson, D. Maltz, and J. Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. In *Internet Draft, draft-ietf-manet-dsr-07.txt*, work in progress, 2002.
7. C.E. Perkins and T.J. Watson. Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM'94 Conference on Communications Architectures*, London, UK, 1994.

Dynamic AODV Backup Routing in Dense Mobile Ad-Hoc Networks*

Wen-Tsuen Chen and Wei-Ting Lee

Department of Computer Science, National Tsing Hua University,
Hsin-Chu, Taiwan 300, ROC

Tel:+886-3-5742896 Fax:+886-3-5711484

wtchen@cs.nthu.edu.tw, leif@mercury.cs.nthu.edu.tw

Abstract. The frequent change of network topology in mobile ad-hoc network leads to the stability and reliability problems of routing. Many routing schemes such as multi-path routing and backup path routing are proposed to increase the link reliability. Multi-path routing protocols usually concentrate on load balancing or disjoint routing. However, the problem of packet loss caused by re-routing from the source to the destination is ignored. In this paper, we propose the Dynamic AODV Backup Routing Protocol (DABR) to enhance the Ad hoc on-Demand Distance Vector (AODV) routing in dense mobile ad-hoc networks. The DABR follows the route discovery mechanism of AODV and dynamically calculates the backup routes. Upon the failure of primary route, data packets can be salvaged by redirecting them to the backup routes. The simulation results show that the link reliability of DABR is higher than the conventional AODV while the overhead is controlled.

1 Introduction

In recent years, mobile ad-hoc networks are applied in more and more areas. The characteristics of ad-hoc networks such as infrastructureless and mobility make it easy to deploy in many areas including academia, business, and military. In mobile ad-hoc networks, nodes are considered to be routers which can forward packets and can move freely within the coverage of network. The movement of node results in the change of network topology and the change of routing. The function of routing protocol is to maintain the correct routes even if the topology changes frequently. Besides the problem of changed topology, ad-hoc routings suffer from many strict problems. In ad-hoc networks, low bandwidth (compared to the bandwidth of wired networks), limited battery life, variable nodal density, and potentially large number of nodes make the routing protocol difficult to design.

Many routing schemes are proposed for ad-hoc networks. They can be classified into two catalogs, on-demand and proactive routing. The former includes AODV [1], DSR [2], ABR [6], and etc. The latter includes DSDV [5], OLSR [4], FSR [3] and etc.

* This work was supported by the Ministry of Education, Taiwan, R.O.C. under Grant 89-E-FA04-1-4.

In the fashion of on-demand scheme, the source node discovers a route to the destination node only when the route is needed. Mobile nodes usually follow the Route Request (RREQ) / Route Reply (RREP) mechanism to discover a route dynamically. For example, when the source node wants to communicate with the destination node but the route is unknown, the source will broadcast a RREQ message to the networks. The RREQ will be propagated throughout the network until it is received by the destination or is intercepted by an intermediate node which knows a route to the destination. Then a RREP message is replied to the source in the form of unicast and the route is established. The behavior described above is called the route discovery. The main advantage of on-demand routing protocol is that it won't incur any control overhead when there are not any communications in the network. Hence, the change of topology only affects the active routes.

In the other hand, the proactive routing protocols calculate the routes to every node proactively based on the global network information. Each node should periodically or triggered broadcast its routing information (e.g., link-state or distance vector) through the entire network. According to the collected routing information, the node produces a routing table which contains routes to every reachable node. The advantages of proactive routing are low latency of routing setup, good resilience of re-routing, and high capability of route status monitoring. However, the proactive routing suffers from many problems. If the number of nodes increases dramatically, the exchange of routing information will incur very large overhead. And the size of routing table is proportioned to the number of nodes in the network. That is, the demand of both storage and computation capability will increase as the network scale grows.

In order to provide more route reliability onto the on-demand routing, many approaches are proposed to find multiple paths [12], [13], [14] rather than just one shortest path. Multi-path routing schemes allocate multiple paths at the phase of route discovery and deliver data packets among these paths to balance the load of traffic. If one of the paths fails, the source node can use the other path(s) to deliver data packets. Although the reliability of path is increased and the delay of reconstructing a new route is eliminated, the data packets which are sent onto the failure path are missing. Packet loss is not handled in MAC or IP layer but is expected to be recovered in higher layer such as TCP or application layer. Even though all the missing packets can be recovered, the end-to-end delay is produced.

The backup path routing is another type of multiple path routings. Multiple short backup routes are attached to the active primary route [7]-[9]. Data packets are delivered along the primary route rather than distributed them among the backup routes. In general, mobile nodes in the primary route should exchange the routing information with their neighbor nodes [8], [9]. Therefore, the scope of backup path is limited to the vicinity and the length of back path is also restricted. When the primary route is disconnected (due to the absence of relay nodes or radio shadowing), the data packets which are on transmission can be salvaged by re-directing them into the backup route without any delay.

An ideal backup routing protocol in on-demand fashion should achieve the goals as follows.

- (1) High delivery rate and low loss rate of data packets
- (2) Transparent to the source node
- (3) Correct and loop-free backup routes establishing

(4) Precise lifetime (which is the period between the creation and the destruction) of backup routes

(5) Low overhead of maintaining the backup routes

In this paper, we investigate the issue of backup routing which is based on the on-demand routing protocol in the environment of dense mobile ad-hoc networks. We propose a Dynamic AODV Backup Routing (DABR) protocol to dynamically build the backup routes with low control overhead. The DABR follows the standard RREQ/RREP messages of AODV and introduces two new message types: *Alternative Route Request (AREQ)* and *Alternative Route Reply (AREP)*. In DABR, the finding of backup route is initiated after the establishment of primary route and is invisible to the originating node. In order to avoid incurring too much overhead, the length of backup routes is limited.

The rest of this paper is organized as follows. Section 2 states the previous works of backup routing protocols. Section 3 describes the details of DABR protocol. Section 4 shows and explains the simulation results and Section 5 concludes the paper.

2 Related Works

Several previous works has been proposed to enhance the link reliability and lower the packet loss in the network layer. Both AODV and DSR have their own mechanisms to salvage the data packets. And several derived approaches have also been proposed.

The local repair mechanism of AODV in [1] is defined as an option. If a link breaks, the upstream node of the broken link can repair the link by initiating a RREQ for the destination and waiting for a RREP. The flooding range of the RREQ is restricted by setting the TTL and the value must be shorter than the formal one. The limited TTL prevents the node from selecting a backup path which is too long. During the local repair, data packets are buffered and thus the end-to-end delay occurs here. If the node receives a RREP before the timeout, the alternate route is set and the data packets are sent to the route. Otherwise, the buffered data packets will be discarded and the RERR message is delivered for the destination. Note that there may be not any overlap between the alternate path and the original path. That is, the backup path may quite different from the original path.

The broken link can be repaired proactively before the incoming packets suffer the transmission error. If the MAC layer could provide the notification of link error to the network layer, the route can be repaired earlier. As soon as the link fails, the incoming packets can be forwarded to the backup route without any delay. However, that the routes which are no longer in active still may be repaired will consume the bandwidth of network.

The AODV-BR [7] is proposed to provide AODV with backup routing without producing any control messages. The alternative paths are set during the propagation of RREPs. Every node must overhear the RREPs which are sent to its neighbors and store the senders in the alternate routing table. After the propagation of RREPs from the destination to the source, the primary and alternate routes will form a fish bone structure, which is illustrated in Figure 1. If one link in the primary route fails, the upstream node of the broken link must broadcast the data packets to its neighbors and

issue a RERR message to the originator. This behavior causes that the source node starts the RREQ/RREP mechanism again to discover a new route for the destination. In the same time, however the broadcasted packets may be salvaged if the neighbors know the alternate route to the destination.

The AODV-BR is based on two characters. One character is that every node can overhear the RREPs. To do this, the node must receive all packets regardless of the destination. The other character is that the node must broadcast the data packets to salvage them if the local link breaks. However, all the nodes which know the alternate route to the destination will forward the data packets. Multiple copies of the data packets will consume the bandwidth and the destination node will receive the redundant packets. Although the AODV-BR claims that it doesn't produce any additional control overhead, the two characters make the protocol difficult to implement in reality. Additionally, the behavior of AODV-BR will be in vain if the topology of neighbors changes after the setup of primary route.

Neighborhood Aware Source Routing (NSR) [8] is proposed to improve the capability of backup routing in DSR. In NSR, both the nodes in the primary route and their 1-hop neighbors should broadcast their 1-hop link-state information. Therefore, every node in the primary route maintains the network topology within two hops. Consequently, the backup routes or shortcuts are computed by the partial topology dynamically by comparing the source routes in the Route Cache and the partial topology. If a link breaks, the upstream node of the broken link must replace the original source route on the packet with the backup route which has been calculated proactively. NSR utilizes the node id to lower the size of control messages and simplify the computation of backup routes. However, the Route Error message should be delivered to notify the other nodes. The reason is that multiple portions of backup routes make the long path. The long path results in the long round trip time. Therefore, the source node should discover the new route for the destination even if there are backup routes.

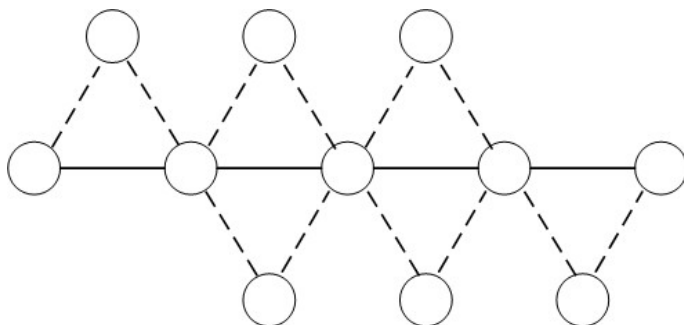


Fig. 1. The primary path and the backup paths

3 Dynamic AODV Backup Routing (DABR) Protocol

The DABR protocol focuses on dynamically finding the backup routes for the existing primary route, which is built by the route discovery mechanism of AODV.

Besides the normal routing table, every node must maintain an additional one, called the backup routing table. The backup routes can not exist without the primary route. In fact, the life of backup routes begins at the establishment of the primary route and ends when the given lifetime is up.

Figure 2 shows the finite state machine of backup routing. When the MAC layer detects the occurrence of link failure, the state is changed from normal route to error route. The RERR is immediately delivered to announce that the link breaks here and every node should not use this link anymore. If the backup route exists, the state transfers to the backup route. In this state, the incoming data will be salvaged by redirecting them into the backup route rather than discarded or buffered. However, once the new route has learned from the receiving of RREP, the state enters the normal route. In the state of normal route, every node uses the normal route to deliver and forward packets.

Nodes in the primary route must notify its neighbors that the backup routes are needed by broadcasting the *AREQ* messages containing the routing information, named vector here. The propagation range of *AREQ* is limited to control the overhead. According to the collected vectors in the *AREQs* from other nodes, the neighbors can determine whether a backup route exists. If there are backup routes, the neighbor will reply *AREPs* to the nodes in the primary route.

The DABR uses three types of message to discover backup routes. They are listed and described in Table 1.

3.1 Message Formats

In this section, we define the format of messages used in DABR, including *AREQ*, *AREP* and *AERR*. The format of *AREQ* is defined in the following.

***AREQ* = <Source Address (srcAddr), AR Sequence (arSeq), Previous Address (preAddr), Hop Count, Vector>**

The first field is the address of the node which originates the message. The **arSeq** is the sequence number binding to the *AREQ* message. The **preAddr** is the address of the previous node which initiates or forwards the *AREQ* to the receiving node. The hop count from the source to the node receiving the *AREQ* will be saved in the **Hop Count** field. The last field **Vector** contains the routing information of the source node. The backup routes are computed by the collected **Vectors**. Its format is defined in the following.

Vector = {Terminus Address (tmAddr), Terminus Sequence (tmSeq), Hop Count to Terminus (HC2T), Source Address (srcAddr), Lifetime}

The **Vector** contains the routing information of the source node which initiates the *AREQ*. The **tmAddr** is the address of the destination node which the backup routes for it are requested. The **tmSeq** is the sequence number binding to the terminus node. The **HC2T** is the hop count from the source node to the terminus node. The hop count represents the position of node in the primary route. **HC2T** is the main criterion to

determine the backup routes. Additionally, the **Lifetime** of vector is given to control the lifetime of vectors. If the lifetime is expired, the vector will be deleted. Next, we define the format of *AREP*.

***AREP* = <Terminus Address (tmAddr), Hop Count to Terminus (HC2T), Previous Address (preAddr), Lifetime>**

The **tmAddr** of *AREP* is the address of the destination node. The **HC2T** is the distance from the node sending the *AREP* to the terminus node. The node in the primary route uses the value to determine a shortest backup route if receiving multiple *AREPs*. The **preAddr** is the address of the node which initiates or forwards the *AREP*. The **Lifetime** field is also required to limit the lifetime of backup routes. When the lifetime is up, the backup route is removed from the backup routing table.

***AERR* = <Terminus Address (tmAddr), Terminus Sequence (tmSeq), HC2T>**

The *AERR* is defined in the above. The meaning of each field is the same as that have mentioned in the previous paragraphs.

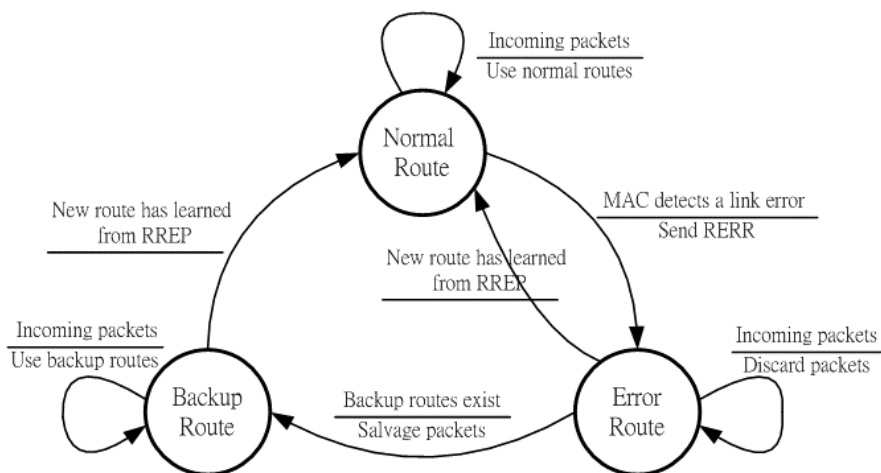


Fig. 2. The finite state machine of backup routing

3.2 Operations of DABR

In AODV routing protocol, both RREQ and RREP messages have the field of hop count. It is easy to record the number of hops from the source to the node which receives the message. Therefore, in the phase of route discovery, the node can obtain the information of hop count to the two termini.

Table 1. The message types of DABR

Message	Abbreviation	Description
Alternative Route Request	AREQ	Nodes in the primary route broadcast the <i>AREQs</i> to notify their neighbors that the backup routes are needed.
Alternative Route Reply	AREP	The neighbors of the primary route send <i>AREPs</i> to show the existing of the backup routes
Alternative Route Error	AERR	The node which encounters the broken link from the backup route should send <i>AERRs</i> to notify other nodes.

Thus, the **Vector** can be formed. The receiving of RREP implies that the primary route is established. Subsequently, the node starts broadcasting *AREQs* to its neighbors periodically. The **HC2T** in the sending *AREQ* is copied from the local saved **HC2T**. When initiating a new *AREQ*, the node must increase its own **arSeq**. The pair of **arSeq** and **srcAddr** can determine a unique *AREQ*.

3.2.1 Operation of Receiving an AREQ

When a node receives the first *AREQ*, it does not react immediately but waits for a timeout. If the *AREQ* comes from the immediate upstream or downstream node, the message should be discarded. During the timeout, the node collects the *AREQs* and caches the **Vectors** contained in the messages. After the timeout, the node finds the nodes which **HC2T** are smaller than the node itself according to the **Vectors**. These nodes with smaller **HC2T** are candidates of the backup next hop. In order to keep the maintenance simple, each node only maintains one backup next hop for each destination. Therefore, the backup next hop is set to the node with smallest **HC2T** among the candidates. Next, the node should send *AREPs* to notify its possible immediate upstream nodes that there is a backup route to the destination. The sending *AREP* carries the **HC2T** of the backup next hop plus one.

As shown in Figure 3, for example, the path $s \rightarrow a \rightarrow b \rightarrow c \rightarrow d$ is the primary route and the two termini are s and d . In this case, node f can hear the *AREQs* from node a , b and c . The corresponding **Vectors** are $\langle 7, a \rangle$, $\langle 6, b \rangle$ and $\langle 5, c \rangle$ (the first field represents the **HC2T** to the terminus d and the second field is the id of nodes). Since node f is not in the primary route, it doesn't know the **HC2T** to d . Node f just select node c as the backup next hop because the **HC2T** of c is the smallest one. Then, f sends *AREPs* to the other neighbors which have larger **HC2T** (i.e., node a and b).

If the topology of the primary route changed because of the moving of nodes, the potential shortcuts may exist. In Figure 4, node a receives the *AREQ* from c and the **HC2T** of c is smaller than node a itself. Therefore, node a should select c as the backup next hop for the terminus d . Node a should not send any *AREPs* if there are not any other neighbors having larger **HC2T** than a .

Multiple primary routes could share some portion of common routes and neighbors. The common routes result from that the RREQ is replied by the

intermediate node rather than the destination node. Figure 5 shows that the two primary paths $s \rightarrow a \rightarrow b \rightarrow c \rightarrow d$ and $e \rightarrow f \rightarrow g \rightarrow c \rightarrow d$ have the common route $c \rightarrow d$ and the common neighbor h and i . Namely, h can receive *AREQs* for the same terminus d from node a, b, f and g while i can hear *AREQs* from b and g . In the view of h , both b and g have the smallest **HC2Ts** but only one should be selected to be the backup next hop. What node should h choose depends on the order of receiving the *AREQs*. Suppose that the *AREQ* from b comes more early than g , node h chooses b as the backup next hop. Next, h should decide what upstream nodes should be notified by the *AREPs*. Now, the **HC2T** of backup next hop is 5 and therefore h should reply the *AREPs* to a and f because their **HC2Ts** are larger than 5.

The routing loop may be produced when the link $b \rightarrow c$ and $g \rightarrow c$ break at the same time. If h sends *AREP* to g , and i sends *AREP* to b , the routing loop $g \rightarrow h \rightarrow b \rightarrow i \rightarrow g$ is formed. In order to avoid the problem, h must not send *AREP* to g . Similarly, node i should not send any *AREPs* because the **HC2Ts** of b and g are tied. The policy is that the node should not send any *AREPs* to the nodes which have the same **HC2Ts** as the backup next hop.

3.2.2 Operation of Receiving an AREP

When a node receives an *AREP* message, it first checks whether there is already a backup route to the **tmAddr**. If there are not any backup routes, the node assigns the node which initiates the *AREP* to be the backup next hop. Otherwise, the node will choose the shorter route by comparing the **HC2Ts** in the backup routing table and the *AREP*.

For example, when node a in Figure 3 receives the *AREP* from e , node a should set e as the backup next hop for the terminus d . Subsequently, if node a hears the *AREP* from f , node f will substitute for node e as the backup next hop because the **HC2T** of f is smaller than e . The situation of node b is somewhat different from a . Node b hears the *AREPs* from f and g and they have the same **HC2Ts**. If the *AREP* from f comes more early than g , node b will select f as the backup next hop and discard the *AREP* from g .

3.3 Maintenance of Backup Routes

When the backup route fails, the upstream node of the broken link should send *AERR* to claim that the other nodes should not use the broken link anymore. Actually, the function of *AERR* is the same as the *RERR* message. However, unlike the broadcasting of *RERR*, the TTL of *AERR* is limited to one or two hops (the value is corresponding to the TTL in *AREQs*).

To avoid the heavy control overhead, the maintenance of backup routing table is based on the lifetime control. Both **Vectors** and backup routing entries have lifetimes. The expired **Vectors** are deleted from the cache and the expired backup routes are marked as inactivated.

3.4 Range Extension of AREQ

The propagation range of *AREQs* can be extended to two hops by controlling the **Hop Count** field. Every node in the primary route broadcasts *AREQs* to at most 2-hop away. The **HC2T** plus the **Hop Count** in the **Vectors** is used to determine the shorter route. For example in Figure 6, node *f* hears *AREQs* from *a*, *b* and *c*. Node *f* will select *c* as the backup next hop because the value of **HC2T** plus **Hop Count** in the **Vector** is the smallest one. Subsequently, it will send *AREPs* to *e* and *b* since their **HC2T** plus **Hop Count** are larger than *c*. The range extension of *AREQ* will broaden the length of backup routes. However, if the total length of the backup routes is too large, the delay of data packets will increase.

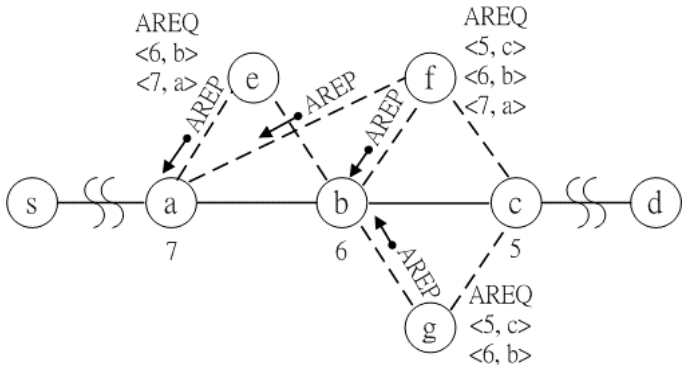


Fig. 3. The primary route and backup routes in the DARR

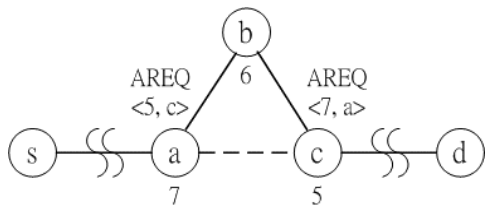


Fig. 4. The shortcut in the primary route

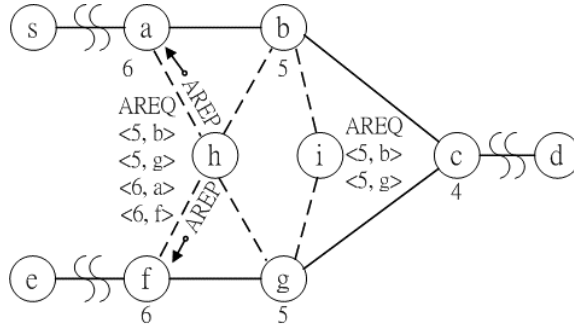


Fig. 5. The overlapping neighbor nodes between multiple primary routes

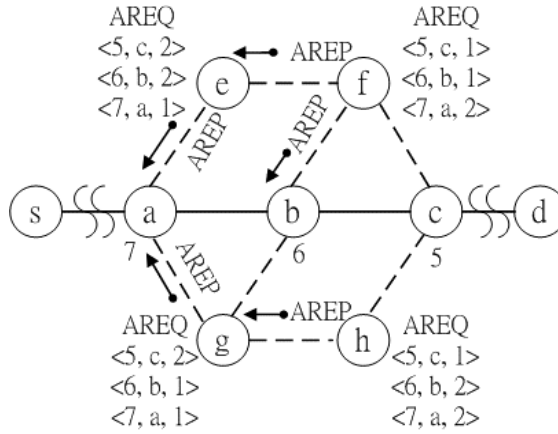


Fig. 6. Backup Routes with 2-hop AREQs

4 Performance Evaluation

In order to evaluate the improvement of performance made by the DABR protocol, we compare the AODV and the DABR via simulation.

4.1 Simulation Environment

The simulation is based on the GloMoSim [10] which is a network simulation library built by the PARSEC [11]. The PARSEC is a language for parallel execution of discrete-event simulation.

Initially, mobile nodes are uniform distributed within the simulated terrain of 1000 * 600 meter². The 802.11 MAC protocol is utilized and the transmission range of nodes is about 180 meters. The mobility model is the random way-point [2], i.e.,

every node selects a random target location and moves to the target with a fixed speed. After arriving at the target, the node pauses for a period of time and randomly selects the next target to move to. In this simulation, the pause time of nodes in all experiments is set to zero.

In order to focus on the effects of routing protocols in network layer, the traffic pattern of application layer in simulation is CBR (Constant Bit Rate) under UDP. When the simulation starts, CBR clients and CBR servers are randomly assigned and will not change through the experiment. The connections will last till the end of simulation. The item size of CBR is 512 Bytes and the time intervals between the items to be sent are from 0.5 to 0.9 second. The delivery rate of IP packets is measured by counting the sending and receiving items. The ratio of all received packets in destination nodes over all sending packets of source nodes is the delivery rate of data packets. We also measure the control overhead by counting the number of control packets which are delivered from all nodes in the network. Besides the control packets of AODV (i.e., RREQ/RREP/RERR), control packets of DABR include AREQ and AREP. The propagation range of both AREQ and AREP is 1-hop. The simulated time in all experiments is 10 minutes.

4.2 Simulation Results

The simulation results in Figure 7 show that the delivery rate of data packets in DABR is higher than AODV. The IP packets will be dropped if there are neither normal routes nor backup routes. The different speed of nodes is specified in the x-axis. When the nodal mobility is high (i.e., the speed of nodes is high), the improvement of DABR is excellent.

The control overhead is the number of control packets sent by nodes in the network. As shown in the Figure 8, the DABR incurs more control overhead than AODV, however, the additional overhead is almost constant. Although the data packets which encounter the broken link are salvaged by redirecting them to the backup routes, the RERR messages should be sent back to the source node in the same time. The source node should discover the newest routes to avoid the large length of backup routes.

5 Conclusion

In this paper, we develop the DABR protocol to enable backup routing in the AODV. The simulation results show that the additional overhead of DABR is almost constant. However, the gain of packet delivery rate grows with the nodal mobility. We conclude that the DABR is a simple and overhead controlled backup routing protocol which outperforms the AODV in link reliability, especially in the network with high nodal mobility.

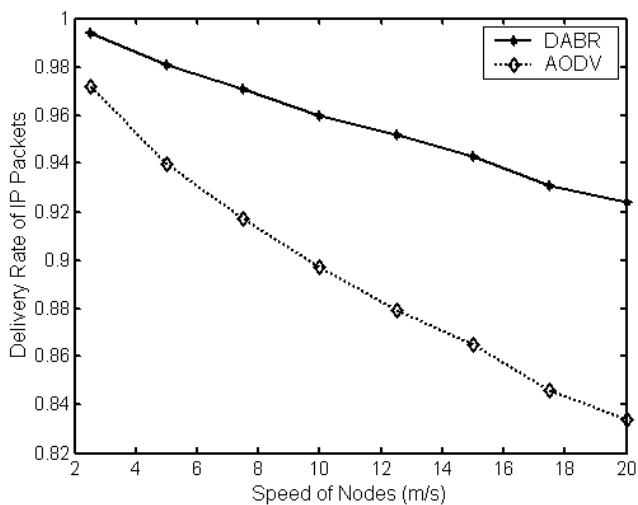


Fig. 7. The delivery rate of data packets (5 connections in 60 nodes)

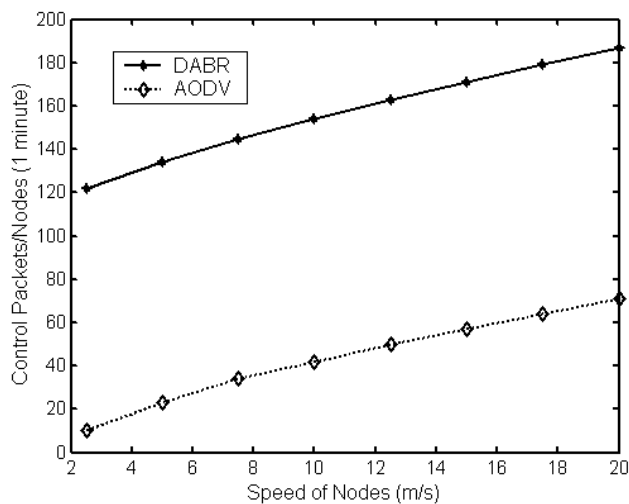


Fig. 8. The control overhead per nodes in one minute (5 connections in 60 nodes)

References

1. C. E. Perkins, E. M. Belding-Royer and S. R. Das, "Ad hoc on-Demand Distance Vector (AODV) Routing," *Internet Draft* (work in progress), draft-ietf-manet-aodv-12.txt, 4 Nov. 2002.
2. D. B. Johnson, D. A. Maltz, Yih-Chun Hu and J. G. Jetcheva, "Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," *Internet Draft* (work in progress), draft-ietf-manet-dsr-07.txt, Feb 2002.
3. G. Pei, M. Gerla, and T.-W. Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks," in *Proc. IEEE ICC 2000*, vol. 1, 2000, pp. 70–74.
4. T. Clausen et al, "Optimized Link State Routing Protocol," *Internet Draft*, (work in progress) draft-ietf-manet-olsr-07.txt, Dec. 2002.
5. C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *Proc. ACM SIGCOMM'94*, vol. 24, Oct. 1994, pp. 234–244.
6. C.-K. Toh, "Associativity-Based Routing For Ad Hoc Mobile Networks," *Wireless Personal Communications Journal, Special Issue on Mobile Net-working and Computing Systems*, Kluwer Academic Publishers, vol. 4, no. 2, Mar. 1997, pp. 103–139.
7. S.-J Lee and M. Gerla, "AODV-BR: Backup Routing in Ad hoc network," in *Proc. IEEE WCNC 2000*, vol. 3, 2000, pp. 1311–1316
8. M. Spohn and J. J. Garcia-Luna-Aceves, "Neighborhood Aware Source Routing," in *Proc. ACM MobiHoc 2001*, 2001, pp. 11–21.
9. W.-P Chen and J. C. Hou, "Dynamic, Ad-hoc Source Routing with Connection-Aware Link-State Exchange and Differentiation," in *Proc. IEEE Globecom'02*, vol. 1, 2002, pp. 188–194.
10. X. Zeng, R. Bagrodia and M. Gerla, "GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks," in *Proc. IEEE PADS 98*, May 1998, pp. 154–161.
11. R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y. Song, "PARSEC: A Parallel Simulation Environment for Complex Systems", *IEEE Computer*, vol. 31, Oct. 1998, pp. 77–85.
12. S. -J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," in *Proc. IEEE ICC 2001*, vol. 10, 2001, pp. 3201–3205.
13. M. R. Pearlman and et al, "On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Network," in *Proc. ACM MobiHoc 2000*, 2000, pp. 3–10.
14. P. Pham and P. Perreau, "Multi-path Routing Protocol with Load Balancing Policy in Mobile Ad Hoc Network," in *Proc. IEEE MWCN 2002*, 2002, pp. 48–52.

Multipath Power Sensitive Routing Protocol for Mobile Ad Hoc Networks

Anand Prabhu Subramanian, A.J. Anto, Janani Vasudevan, and P. Narayanasamy

Department of Computer Science and Engineering
Anna University, Chennai – 600 025, India.
{anand_ps2000, jesusanto, jananivasu}@yahoo.com
sam@annauniv.edu

Abstract. Mobile Ad hoc Networks are characterized by multi-hop wireless links, without any infrastructure, and frequent host mobility. A plethora of routing protocols has been proposed. A class of routing protocols called *on-demand* protocols has recently gained attention because of their efficiency and low routing overhead. As the mobile nodes in the network work on low power batteries, the need to take into account their *power consumption* arises. This paper focuses on a particular on-demand routing protocol, called *Dynamic Source Routing*, and shows how an efficient heuristic based Multipath technique can improve the mean time to node failure and maintain the variance in the power of all the nodes as low as possible. In the *Multipath Power Sensitive Routing Protocol* (MPSR) every node in the network is treated equally and the overall network is stable for a long time. An interesting feature of using this protocol is that the end-to-end packet delay does not increase significantly. The results of extensive simulation show that the performance of *MPSR* protocol is on an increasing trend as mobility increases when compared to the *Dynamic Source Routing*.

1 Introduction

Mobile hosts and wireless networking hardware are becoming widely available, and extensive work has been done recently in integrating these elements into traditional networks. However, mobile users may want to communicate in situations in which no fixed wired infrastructure is available, because either it may not be economically practical or physically possible to provide the necessary infrastructure or because the expediency of the situation does not permit its installation. A mobile ad hoc network is an autonomous system of mobile hosts connected by wireless links. There is no static infrastructure such as base stations. If two hosts are not within radio range, all message communication between them must pass through one or more intermediate hosts that act as routers. The hosts are free to move around randomly, thus changing the network topology dynamically.

The design of an efficient routing protocol is a major challenge in such dynamic wireless networks. A lot of work has been done in this area right from the seventies, when the U.S. Defense Research Agency, DARPA supported the PRNET (Packet Radio Network) [7]. Routing protocols must deal with the typical limitations of these

networks that include high power consumption, low bandwidth and high error rates. These routing protocols may generally be categorized as *table driven* and *source initiated on demand driven*.

On-demand routing is the most popular approach in ad hoc networks. Instead of periodically exchanging route messages to maintain a permanent route table of the full topology, on-demand routing protocols build routes only when a node needs to send data packets to a destination. Most proposed protocols of this type (for example, Dynamic Source Routing (DSR) [3] and Ad hoc On-demand Distance Vector (AODV) [2]) however, use a single route for each session.

Multiple paths can be useful in improving the effective bandwidth of communication, responding to congestion and heavy traffic, and increasing delivery reliability. Multipath routing protocols in wired networks has been widely developed in [9], [6], [4], and [19]. These protocols use table-driven algorithms (link state or distance vector) to compute multiple routes. Studies show however, that proactive protocols perform poorly in mobile networks because of excessive routing overhead [8], [13]. Multipath routing in ad hoc networks has been proposed in [16], [1], [21], and [10]. Although these protocols build multiple routes on demand they are not much concerned with the power of each node in the route. Providing multiple routes is beneficial in network communications, particularly in mobile wireless networks where routes become obsolete frequently because of mobility and poor wireless link quality.

We approach to find more than one efficient path between a source and destination to mask link failures in the network. This requires three components: A route discovery mechanism, a mechanism for sending packets along the selected route and a high level protocol for selecting the most reliable set of routes from the many paths that may exist in the route cache of the source node. The first two of these, discovery and forwarding mechanisms, are relatively well-understood. In order to accomplish the third issue, we use a heuristic to find which of the potentially efficient routes in the cache the routing layer should use to achieve high network stability and maintain the variance in power of the nodes to a minimum.

This paper presents Multipath Power Sensitive Routing (MPSR) Protocol that builds multiple routes between the source and the destination nodes and uses a heuristic to switch between routes such that the burden of routing is distributed evenly to all the nodes in the network. The authors believe that every node in the network must be treated equally for the stability of the network. This paper aims to achieve this stability by maintaining the variance of the remaining power of each node as low as possible so that the mean time to failure of the nodes increases.

The remainder of this paper is organized as follows. In Section 2 we survey the applied work on Multipath routing. Section 3 briefly describes the DSR protocol and the circumstances in which its performance is not satisfactory. Section 4 describes the proposed Multipath Power Sensitive Routing (MPSR) Protocol in detail. Performance evaluation by extensive simulation is described in Section 5. Section 6 describes the future work intended to be performed. Section 7 concludes the paper.

2 Related Work

Past work on multipath routing protocols has mainly focused on quick failure recovery, finding disjoint paths between source and destination. Here we approach to use a heuristic (a combined metric using the shortest path and the average remaining power of the nodes in the route) to effectively select multiple paths.

In On-demand multipath routing for mobile adhoc networks [1] the route requests that are replied to are those that carry a source route that is link wise disjoint from the primary source route. When the primary route breaks, the shortest remaining alternate route is used. It shows that providing all intermediate nodes in the primary (shortest) route with alternative paths has a significantly better performance than providing only the source with alternate paths.

The AODV-BR: Backup routing [16] algorithm discovers more than one route in order to replace a broken one with one of the backup routes. It relies on variants of the on-demand routing protocol, AODV, to discover multiple routes. The goal is to improve the packet delivery ratio and the average delay per packet by falling back to an operational backup route when the primary route breaks.

Split Multipath Routing with Maximally Disjoint Paths [17] approaches to use both primary and backup paths simultaneously to route data. Such a multipath routing approach can better distribute load, resulting in significant decreases in packet loss and, in the case that packets are dispersed across the path set with increased fault tolerance. [17] has examined how to establish two maximally disjoint paths and select routes on a per-packet basis. These protocols do not address the issue of path selection and are limited to route replies provided by the routing protocol, and [1] does not provide a specific method for selecting the maximally disjoint path.

The Path Set Selection [14] deals with finding redundant paths that are disjoint in nature. This is to ensure that the correlation between the failures of the paths is as small as possible. The Alternate Path Routing [12] selects the two routes with the least number of hops after decomposing the route replies into constituent links. Further more this protocol does not provide a metric to justify route selection scheme.

Here we take care that the route replies are sent to the source node in such a way that the efficient routes reach the source in order of their efficiency. We define the efficiency of the routes by using the average remaining power of the nodes in the route and hop count so that the end-to-end delay does not increase significantly. The forwarding function chooses the currently available efficient route and sends packets through this route.

3 Dynamic Source Routing Protocol

In the Dynamic Source Routing (DSR) Protocol, a process known as *Route Discovery* establishes route between a source and a destination node. Here the source floods a route request packet and as the route request packet moves towards the destination node, the route gets built. When the route request packet reaches the destination it sends the route reply packet to the source. Any intermediate node knowing the route to the destination can also send a route reply. After knowing the route the source adds the route to the route cache it maintains and also to each data packet that it wants to

send to that particular destination. The route discovery process is done on-demand. Even though many route requests reach the destination it sends the reply packet corresponding to the shortest route.

The link break is taken care by the *Route Maintenance* process, which sends a route error packet to the source saying that a particular link is not valid. The source then initiates a route request if it has more data to send to that destination.

3.1 Circumstances in Which Performance of DSR Is Not Satisfactory

As DSR uses the shortest path as the routing criteria, it may overload some of the nodes that occur in more number of shortest paths. Consider the situation as in Fig 1.

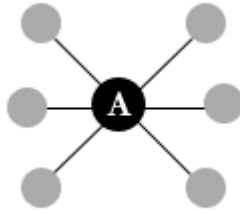


Fig. 1. A network illustrating the situation in which shortest path routing performs poorly

Consider node A that is present in more than one shortest route of different source destination pairs. As it is overloaded to carry many of the data packets, the power of this node decreases more rapidly than other nodes. If node A fails due to power loss, then all the routes that pass through node A become invalid which results in a series of route discovery process. This results in a significant routing overhead in the network and increases the traffic. The aim of this paper is to reduce this type of overhead by associating a weight to each node and making the routing decision based on the weight as well as the shortest path so that it does not significantly affect the end-to-end delay.

Link breaks can be due to mobility or node failure resulting from power loss. As DSR caches only the shortest path, it needs to initiate a route request once a cached route becomes invalid. As mobility increases or when the power of the nodes in the route decreases node failures occur more frequently and the route request packets adds significantly to the routing overhead. When the source nodes cache multiple routes, it can switch over to another cached route so that the time for route discovery is eliminated and network overhead due to route request is reduced. It also ensures that the multiple routes cached in the route cache are not stale by maintaining a time out value for each entry.

These facts motivated us towards a heuristic based multipath power sensitive routing protocol that performs well in the above-mentioned volatile circumstances.

4 Multipath Power Sensitive Routing Protocol

Multipath Power Sensitive Routing Protocol (MPSR) is an on-demand source routing protocol that facilitates the source node to accept multiple route replies. The source can cache these multiple routes and switch between them based on the network conditions. The routing function constructs the routing table based on the weight (remaining power) of the nodes in the network and the forwarding function selects the routes based on a heuristic that aims at reducing the power consumption of the nodes in the route. Here we assume a non-hostile environment in which the nodes relay their original weights. A counter called *route count* (*rCount*) is maintained. This specifies the maximum number of routes that can be cached for a particular destination. *rCount* can preferably have values such as 3, 4, 5 etc based on our simulations. These values were chosen considering the number of nodes in the network and the diameter of the network.

The value of *rCount* is known to all the nodes and is used in the route discovery process. Any node can have at most *rCount* routes for a particular destination. Whenever a source node needs to send a data packet to a destination node it searches its route cache for a route to that particular destination. If the route does not exist, it initiates a route discovery process.

In the route discovery process the source node floods the *route request* packet. The weight of the nodes in the route is added along with their address. The weight of the nodes is its current *remaining power*. As each node relays the *route request* it appends its address and weight. When the *route request* reaches the destination node it adds its address and weight and sends the *route reply* packet. Any intermediate node that has a route to the destination can also send the route reply to the source. Figure 2 shows the route with address and remaining power.

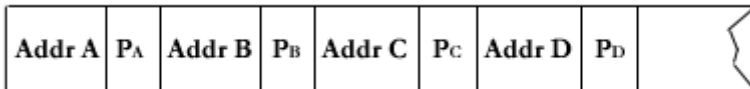


Fig. 2. Address and Power of the nodes in the route request/reply packets

In the destination node the average power of each path is calculated. Let N be the hop count of a path and n be the average power of the path. According to the MPSR routing criteria, the source chooses the path with *high* average power and *low* hop count. So when the destination sends the route reply packet it waits for a time that is equivalent to,

$$(N / n) * t \quad (1)$$

where t is a random amount of time. Using this we can find a route with low hop count and high average power though not with both minimum hop count and maximum average power. This metric is used to get a compromise between the hop count and the average remaining power. This ensures that the more efficient paths reach the source before the source gets *rCount* routes so that the source caches only to

a maximum of $rCount$ efficient routes. The intermediate node that sends the route reply also follows this strategy. This strategy serves for two purposes:

- The destination sends the route with low hop and high average power before any other routes to the source.
- To prevent the traffic around the source node due to excessive route replies. As all the nodes other than the source are potential candidates to send the route reply, there may be lots of route reply reaching the source node.

The route cache in each node is modified such that it can cache more than one route for a destination. The cache is maintained as a priority queue with the most efficient path in the front of the queue. When a new route reaches the source, it checks the number of routes with its $rCount$. If it can cache more routes it adds this route according to the priority queue in the correct position. Thus the insertion of route takes about $O(rCount)$ time. As the value of $rCount$ is of the order of 3, 4, 5 etc this does not affect the computational time for route insertion. As the most efficient route is available in the front of the priority queue the route selection can be done in $\Theta(1)$. We maintain a time out interval for each cache entry so as to eliminate stale routes. In our simulations we used the Timer- Based Route Expiry as described in [11], which is a dynamic mechanism that allows each node to choose timeout values independently based on its observed route stability. Moving on to forwarding, the route selection is based on two strategies.

4.1 Min Power Strategy

For each valid route in the cache for a particular destination, the remaining power of each node in the route is known. Let $minPower$ be the power of the node with minimum remaining power in the route. A parameter called the threshold (t_h) is defined as the value of the safe lower bound of the remaining power. Each node must have the remaining power greater than t_h to function properly. Another parameter $diff$ is defined as follows:

$$diff = minpower - t_h \quad (2)$$

The power discharge pattern of alkaline batteries is linear [18] while the lithium ion batteries have a precipitous discharge in battery life which is quadratic in nature [15]. Considering this power discharge pattern of batteries in the nodes and the value of $diff$, the numbers of packets that can be safely transmitted through this route are calculated based on this criterion. Consider this value to be N_p . Here $N_p = diff / power$ for transmitting each packet based on the discharge function of the battery in the node.

4.2 Round Robin Strategy

The source node has a count of the total number of data packets to be sent to a particular destination. Using *Round Robin* scheme, let the number of packets that can be sent through each of the routes be N_r . Let N_t be the total number of data packets to

a destination and *numRoutes* be the current number of routes to that destination present in the route cache. Then,

$$N_r = N_t / \text{numRoutes} \quad (3)$$

Using the Round Robin strategy, we equally distribute the packets among the routes in the cache.

If the number of packets determined by the round robin strategy (N_r) is greater than the number of packets that can be transmitted based on the Min Power Strategy ($N_r > N_p$), then it is not possible to adopt round robin strategy. So the number of packets that can be transmitted through a route is $\min(N_p, N_r)$.

Consider the situation in which $N_p > N_r$: In this case if the min power strategy alone is followed, then it might result in draining the power of the node to the minimum threshold so that the mean time to failure of the node decreases and the probability of link failure increases. So the combined strategy works well in the above-mentioned situations.

In the route maintenance process, a link break is detected and a route error packet is transmitted to the source analogous to the DSR protocol. The routes with the advertised link are deleted from the route cache.

Thus the Multipath Power Sensitive Routing (MPSR) Protocol maintains the variance of the power among the nodes in the network to be as low as possible thereby contributing to the longer lifetime of the network.

5 Performance Evaluation

We used a detailed simulation study to evaluate the effectiveness of the Multipath Power Sensitive Routing Protocol described in the last section. The performance of the MPSR was compared to the base DSR protocol. In the following sub-section, we first describe the simulation environment and the performance metrics used, and then present and analyze the simulation results.

5.1 Simulation Environment

The detailed simulation was done using the Global Mobile Simulator (*GloMoSim*) Library [20]. The network was modeled with 50 mobile hosts placed *randomly* within a terrain area of dimension 2000m x 2000m area. The radio model to transmit and receive packets is RADIO-ACCNOISE which is the standard radio model used. The packet reception model is SNR-BOUNDED where a node receives the signal without error if the *Signal to Noise Ratio* (SNR) is more than a specified threshold value. The radio transmission power is set to 15.0 dBm and the sensitivity of the radio is set to -91.0 dBm. Each node has a channel capacity was 2 Mb/s.

The IEEE 802.11 Distributed Coordination Function (DCF) [5] was used as the medium access control protocol. The *random waypoint* model was adopted as the mobility model. In the random waypoint model, a node randomly selects a destination from the physical terrain. It moves in the direction of the destination in a speed uniformly chosen between a minimum and maximum speed specified. After it reaches

its destination, the node stays there for a time period specified as the pause time. The minimum and the maximum speed were set constant to zero and 10 m/s, respectively. The various mobility scenarios were modeled by using different pause times. If the pause time increases mobility decreases and vice versa.

5.2 Performance Metrics

The metrics used for performance evaluation were: (i) *Standard deviation of the Routing load* which is the deviation in the number of control packets (*route request, route reply, and route error*). (ii) *Packet delivery ratio* — the ratio obtained by dividing the number of data packets correctly received by the destination by the number of data packets originated by the source. (iii) *Standard Deviation of remaining power* of the nodes in the network. (iv) *Average end-to-end delay* of data packets - this includes all possible delays caused by buffering during route discovery, queuing delay at the interface, retransmission delays at the MAC, propagation and transfer times; (v) *Number of packet drops*— the total number packets dropped that includes data as well as control packets. (vi) *Average hop count*—the arithmetic mean of the hop counts of all the routes present in the route cache of all nodes.

5.3 Simulation Results

The measurement of all the above-mentioned parameters was performed once for each of the different mobility scenarios.

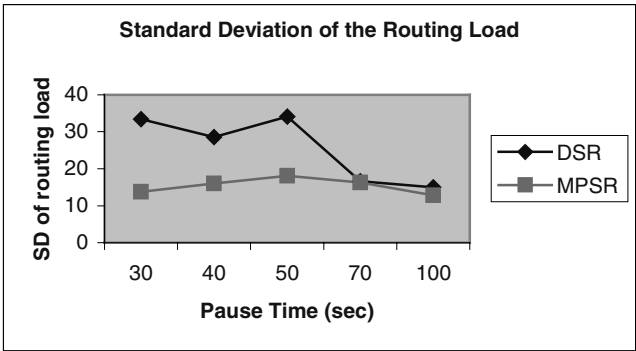


Fig. 3. Plot of the Standard Deviation of the Routing Load with respect to mobility.

Fig. 3 shows the standard deviation (SD) of the routing load which included the control packets viz., route request, route reply and route error. The SD of the routing load in MPSR is observed to be low compared to DSR especially when the mobility is more (when the pause time is less). This is because with high mobility, link breaks are common and DSR needs to initiate new route request and this adds significantly to the routing overhead. Also when the power of the node decreases and when links break

due to link failure, the route error packets are more in DSR. If a link break occurs due to increased mobility, DSR has to initiate a route discovery process. But in MPSR, the probability of all the routes becoming invalid is less and hence the source node can choose another possible route in its route cache to the intended destination. This justifies that there will not be any additional overhead in MPSR when the mobility increases. The low SD of MPSR shows that every node in the network has almost equal routing overhead thus contributing to maintain the SD in the power consumed by the nodes. This increases the mean time to failure of the nodes and the stability of the network.

Fig. 4 shows the packet delivery ratio of each protocol with respect to mobility. The fact that MPSR outperforms DSR is visually obvious, especially when the mobility is higher (i.e., the pause time decreases).

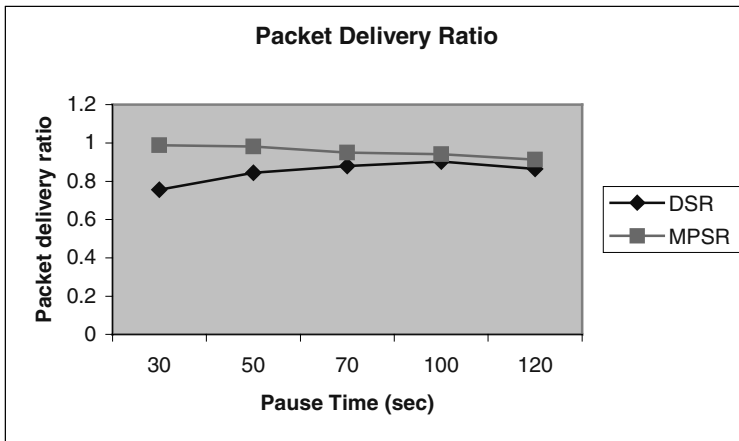


Fig. 4. Plot of the Average Packet Delivery Ratio of nodes with respect to mobility.

In DSR, only one route is used for each session and when that route is invalidated it has no route to that destination. In that case, it sends a RREQ to discover a new route. DSR however, does not apply any aging mechanism for cached route entries, and hence routes stored in the cache (either by the source or the intermediate nodes) may be stale. After a route break, source nodes will use these newly acquired but obsolete routes only to learn that they are also invalid, and will attempt another route discovery. Many data packets are dropped during this process and more delay is needed to discover correct routes.

In MPSR though we maintain multiple routes, a time out interval takes care that the routes are not stale. The performance of MPSR is significant at high mobility.

Fig. 5 shows how the standard deviation in power is lower in MPSR as compared to DSR. The low variance in the control packets is one cause for the low variance in power consumed. Also if the power of a node reaches the lower bound, MPSR routes the packet through other possible safe routes. This ensures that the power consumed by each node is equal. This increases the stability of the network.

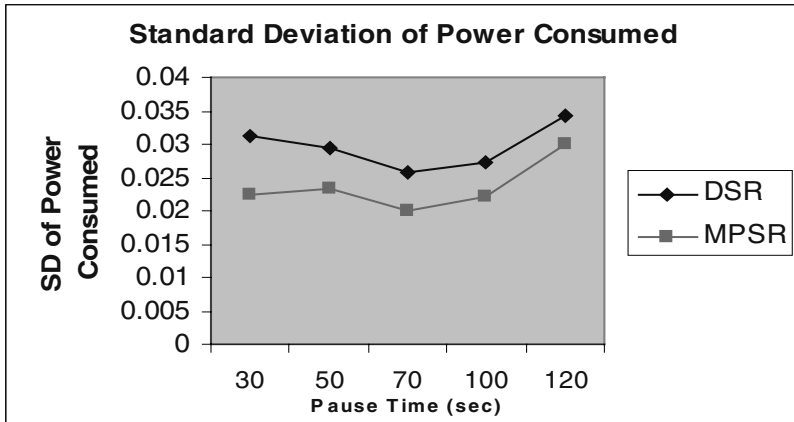


Fig. 5. Plot of the Standard Deviation of the Power Consumed with respect to mobility

The curves for end-to-end delay were not plotted because there was no significant difference in average packet delay between DSR and MPSR. This result was surprising because we had expected a slight worsening in delay for packets (in the MPSR case) as they get routed around nodes with high cost (or high remaining power). This was the result of consideration of the minimum hop (N) and the maximum average power (n) as described in section 3. On closer examination of the simulation trace it was found that some packets did indeed take longer routes and of these some did have higher delay (measured in time steps). However the number of these packets was not large and as a result did not contribute to a statistically significant result. In more congested situations, MPSR performed better than DSR. In addition, DSR yields longer delays reconstructing routes and the period of time the data packets are buffered at the source node during route discovery results in larger end-to-end delays. MPSR on the other hand, uses the remaining valid routes when one of the multiple routes is disconnected, and hence no route acquisition latency is required. So, overall, we conclude that packet delay is unaffected when using MPSR.

Fig. 6 illustrates the number of packets dropped by each protocol. Both data and control packets are measured. The reasons for packet drops can be incorrect route information, mobility, and node failure due to power loss, collisions, and congestion. DSR cannot maintain precise routes and drops more packets as nodes move more often (i.e., less pause time). The usage of stale routes from caches is the major reason of DSR packet drops. MPSR has considerably fewer packet drops compared to DSR. This is because MPSR invokes fewer route discovery processes and consequently, transmits less control.

Fig. 7 reports the average hop count of each protocol. DSR has the shortest hop distance when there is no mobility because MPSR may sometime choose route with longer distance than the shortest route as it considers the remaining power of the

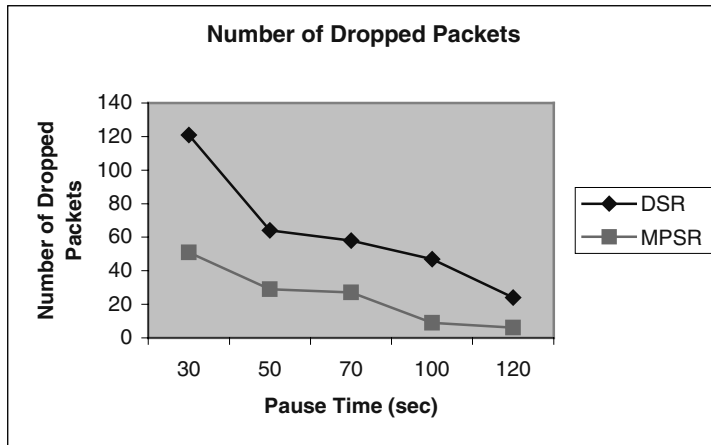


Fig. 6. Plot of the Average number of Dropped Packets with respect to mobility

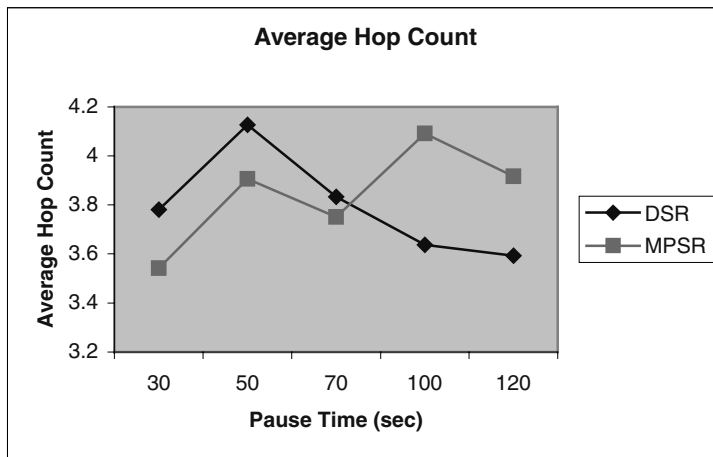


Fig. 7. Plot of the Average Hop Count with respect to mobility

nodes in the path into account. With mobility however, the hop count of DSR grows and becomes larger than those of MPSR protocol. If the route is established directly from the destination, it can be the shortest route since it is built based on the most recent information and accounts for node locations after movements. DSR, however, uses cached routes from intermediate nodes. As there is no timeout interval for the cached routes they may not be fresh and do not exploit the current network topology. DSR therefore builds longer routes than the MPSR protocol. Longer paths have more chance of having route breaks since one-link disconnection results in route invalidation.

6 Future Work

The protocol presented assumes a non-hostile working environment in which the mobile nodes advertise their true weights. We are working on the ways of adapting this protocol in the presence of malicious nodes that advertise their weights lower than what they are, to avoid traffic to pass through them.

7 Conclusion

In this paper, the Multipath Power Sensitive Routing (MPSR) Protocol for Mobile Ad hoc Networks has been presented. MPSR is an on-demand source routing protocol that establishes multiple routes between a source destination pair and switches between the routes based on a heuristic. This heuristic takes the current network conditions into consideration. The remaining power of each node and the hop count are taken as the routing criteria and care is taken such that routing burden is equally distributed over all nodes in the network. This increases the mean time to failure of the nodes and eventually results in the stability of the network. Providing multiple paths is useful in ad hoc networks because when one of the routes is disconnected, the source can simply use other available routes without performing the route discovery process again.

Extensive simulations, which were performed to evaluate the performance of the MPSR and DSR protocols, indicate that MPSR outperforms DSR because multiple routes provide robustness to mobility. The performance difference becomes evident as the degree of mobility increases. MPSR had considerably fewer packet drops compared to DSR. As the packets are transmitted through the route selected based on a combined strategy (*Min Power* and *Round Robin*) we distribute the load evenly to all the nodes and the network is stable for a longer time.

References

1. Nasipuri and S.R. Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks," Proceedings of IEEE ICCCN'99, Boston, MA, Oct. 1999, pp. 64–70.
2. C.E. Perkins and E.M. Royer, "Ad-Hoc On Demand Distance Vector Routing," Proceedings of IEEE WMCSA'99, New Orleans, LA, Feb. 1999, pp. 90–100.
3. D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wire-less Networks," In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, Kluwer Academic Publishers, 1996, pp. 153–181.
4. D. Sidhu, R. Nair, and S. Abdallah, "Finding Disjoint Paths in Networks," Proceedings of ACM SIGCOMM'91, Zurich, Switzerland, Sep. 1991, pp.43–51.
5. IEEE Computer Society LAN MAN Standards Committee, Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.
6. Cidon, R. Rom, and Y. Shavitt, "Analysis of Multi-Path Routing," IEEE/ACM Transactions on Networking, vol. 7, no. 6, Dec. 1999, pp. 885–896.
7. John Jubin and Janet D. Tornow, "The DARPA packet radio network protocols," Proceedings of the IEEE, 75(1):21–32, January 1987.

8. J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," Proceedings of ACM/IEEE MOBIKOM'98, Dallas, TX, Oct. 1998, pp. 85–97.
9. J. Chen, P. Druschel, and D. Subramanian, "An Efficient Multipath Forwarding Method," Proceedings of IEEE INFOCOM'98, San Francisco, CA, Mar. 1998, pp. 1418–1425.
10. J. Raju and J.J. Garcia-Luna-Aceves, "A New Approach to On-demand Loop-Free Multipath Routing," Proceedings of IEEE ICCCN'99, Boston, MA, Oct. 1999, pp. 522–527.
11. M.K Marina, S. R. Das, "Performance of Route Caching Strategies in Dynamic Source Routing," In the Proceedings of 2nd Wireless Networking and Mobile Computing (WNMC), Phoenix, April 2001, In conjunction with the Int'l Conference on Distributed Computing Systems (ICDS) 2001
12. M.R. Pearlman and Z.J. Haas, P. Sholander, S.S. Tabrizi. "On the impact of alternate path routing for load balancing in mobile ad hoc networks." Proceedings of the first workshop on mobile and ad hoc networking and computing (MobiHoc 2000), Boston, MA, Aug. 2000.
13. P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks," Proceedings of ACM/IEEE MOBIKOM'99, Seattle, WA, Aug. 1999, pp. 195–206.
14. P. Papadimitratos, Z. J. Haas, E.G. Sirer, "Path Set Selection in Mobile Ad Hoc Networks," In the Proceedings of the Third ACM Symposium on Mobile AdHoc Networking & Computing (MobiHoc 2002) Lausanne, Switzerland, June 9–11 2002
15. S. Gold, "A PSPICE Macromodel for Lithium-Ion Batteries", The 12th Annual Battery Conference on Applications and Advances, California State Univ., Long Beach, CA, Jan 14–17, 1997
16. S.-J. Lee and M. Gerla, "AODV-BR: Backup Routing in Ad hoc Networks," Proceedings of IEEE WCNC2000, Chicago, IL, Sep. 2000.
17. S.-J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks," Proceedings of ICC 2001, Helsinki, Finland, June 2001.
18. S. Singh, M. Woo and C.S. Raghavendra, "Power Aware Routing in Mobile Adhoc Networks," In the Proceedings of the Fourth Annual International Conference on Mobile Computing and Networking (MOBICOM'98) 1998.
19. S. Vutukury and J.J. Garcia-Luna-Aceves, "An Algorithm for Multipath Computation Using Distance Vectors with Predecessor Information," Proceedings of IEEE ICCCN'99, Boston, MA, Oct. 1999, pp. 534–539.
20. UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems, <http://pcl.cs.ucla.edu/projects/domains/gloimosim.html>.
21. V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," Proceedings of IEEE INFO-COM'97, Kobe, Japan, Apr. 1997, pp. 1405–1413.
22. W.T. Zaumen and J.J. Garcia-Luna-Aceves, "Loop-Free Multipath Routing Using Generalized Diffusing Computations," Proceedings of IEEE INFOCOM'98, San Francisco, CA, Mar. 1998, pp. 1408–1417.

Dependable and Secure Data Storage in Wireless Ad Hoc Networks: An Assessment of DS²

S. Chessa^{1,2}, R. Di Pietro³, and P. Maestrini^{1,2}

¹Dipartimento di Informatica, Università di Pisa,
via Buonarroti 2, 56127 Pisa, Italy

²Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo",
Area della Ricerca CNR, via Moruzzi 1, 56124 Pisa, Italy

³Dipartimento di Informatica, Università di Roma "La Sapienza",
via Salaria, 113 – 00198 Roma, Italy

Abstract. DS² is a dependable and secure data storage for mobile, wireless networks based on a peer-to-peer paradigm. DS² provides support to create and share files under a write-once model, and ensures data confidentiality and dependability by encoding files in a Redundant Residue Number System. The paper analyzes the code efficiency of DS² using a set of moduli allowing for efficient encoding and decoding procedures based on single precision arithmetic, and discusses the security issues. The comparison of DS² with the Information Dispersal Algorithm approach (IDA) shows that DS² features security features which are not provided by IDA, while the two approaches are comparable from the viewpoint of code efficiency and encoding/decoding complexity.

1 Introduction

Mobile ad hoc networks are composed by a set of mobile hosts (also called mobiles) communicating with each other via radio transceivers. In order to communicate with destinations which are located outside of their transmission ranges or hidden by obstacles, communicating mobiles rely on other mobiles which cooperate to forward messages to their destinations. To this purpose the network layer of the mobiles provides services of message delivery by running suitable routing algorithms [1], [2]. However, mobility and failures may give rise to network disconnections impairing service dependability.

Due to mobility of nodes, the network topology varies with time. At a given instant of time it is described by a graph where nodes are the mobiles, and a link connecting two nodes in the graph means that the corresponding mobiles can communicate directly.

The mobiles rely on on-board batteries for energy supply, hence energy efficiency of mobiles is an important issue [3]. The effect of battery depletion is similar to a crash fault, from which the mobile may or may not recover depending on the avail-

ability of battery replacement/recharge. As mobiles may not be equipped with permanent storage, failures may result in data losses or corruption.

An important issue in mobile ad hoc networks is how to implement dependable and secure data storage. This is an essential requirement in applications where the mobiles cooperate by sharing information and need to create and access shared files. The system should prevent data losses or corruption due to network disconnections, mobile failures or malicious attacks from untrustworthy mobiles, and it should provide the file owners with mechanisms for secure distribution of files access privileges.

Several techniques to implement dependable and/or secure data storage have been proposed in the recent literature [4-10]. Some of these, which are based on a client-server paradigm [4], [5], hardly fit the ad hoc network model which is rather based on a peer to peer paradigm. Other approaches are conceived for systems connected with fast, wired networks, where mobility and disconnections of nodes is not supported [6], or which pay considerable communication overhead to implement a sophisticated model of intrusion tolerance based on user authentication [7].

Dependable storage systems based on a peer to peer paradigm which may adapt to the ad hoc network model have also been introduced [8], [9]. They exploit techniques of data fragmentation and dispersal [10] based on erasure codes [11] and use cryptography to achieve data confidentiality.

The technique of data fragmentation and dispersal was originally introduced in [10], where an information dispersal algorithm (IDA) had been proposed. It exploits erasure codes which are optimal with respect to code efficiency and allows for efficient encoding and decoding procedures.

A new technique to achieve dependable and secure data storage (DS^2) in wireless networks has been proposed in [12]. DS^2 exploits *Redundant Residue Number System* (*RRNS*) [13], [14] to encode data, which allows for a uniform coverage of both erasure and errors. *RRNS* encode data as $(h+r)$ -tuples of residues using $h+r$ keys, or moduli. Residues are distributed among the mobiles in the network. Recovering the original information requires the knowledge of at least h residues and of the corresponding moduli. Data can be reconstructed in the presence of up to $s \leq r$ residue losses (erasures), combined with up to $\lfloor \frac{r-s}{2} \rfloor$ corrupted residues. As compared to IDA, DS^2 features basic data confidentiality which is inherently provided by the *RRNS* encoding. Data confidentiality is ensured since mobiles having access to the residues are able to decode them only if they also know the correspondence of the residues with the set of moduli.

In this paper we evaluate the code efficiency and the coding/decoding complexity of DS^2 , we analyze the security issues related to DS^2 and we compare DS^2 and IDA.

The rest of the paper is organized as follows. The Redundant Residue Number Systems and DS^2 are reviewed in sections 2 and 3, respectively. Section 4 and 5 analyze the code efficiency and the encoding/decoding complexity, respectively. Section 6 discusses the security issues and Section 7 compares DS^2 with IDA. Section 8 draws the conclusions.

2 Redundant Residue Number Systems

Given $h+r$ pairwise prime, positive integers m_1, \dots, m_{h+r} called *moduli*, let $M = \prod_{p=r+1}^{h+r} m_p$, $M_R = \prod_{p=1}^r m_p$, and, without loss of generality, $m_p > m_{p-1}$ for each $p \in [2, h]$. Given any non-negative integer X , let $x_p = X \bmod m_p$ be the residue of X modulo m_p . In the rest of the paper also notation $(a)_b$ will also be used to denote $a \bmod b$.

The number system representing integers in $[0, M)$ with the $(h+r)$ -tuples of their residues modulo m_1, \dots, m_{h+r} is called the *Redundant Residue Number System (RRNS)* of moduli m_1, \dots, m_{h+r} , range M and redundancy M_R [13], [14]. For every $h+r$ -tuple (x_1, \dots, x_{h+r}) , the corresponding integer X can be reconstructed by means of the Chinese Remainder Theorem:

$$X = \left(\sum_{p=1, h+r} \left(\frac{M \cdot M_R}{m_p} (x_p \beta_p)_{m_p} \right) \right)_{M \cdot M_R} \quad (1)$$

where, for each $p \in [1, h]$, $\beta_p = \left\langle \frac{M \cdot M_R}{m_p} \right\rangle_{m_p}$ is the multiplicative inverse of $M \cdot M_R / m_p$

modulo m_p , that is, $\left(\frac{M \cdot M_R}{m_p} \beta_p \right)_{m_p} = 1$ [15], and β_p is in the range $[0, m_p)$.

Although the given *RRNS* could provide unique representations to all integers in the range $[0, M \cdot M_R)$ [15], the legitimate range of representation is limited to $[0, M)$, and the corresponding $h+r$ -tuples, are called *legitimate*. Integers in $[M, M \cdot M_R)$ and the corresponding $(h+r)$ -tuples are called *illegitimate*.

Given an *RRNS* of range M and redundancy M_R , with moduli m_1, \dots, m_{h+r} , let (x_1, \dots, x_{h+r}) be the legitimate representation of some X in $[0, M)$. An *erasure* of multiplicity e is an event making unavailable e arbitrary digits in the representation, and an *error* of multiplicity d is an event transforming d arbitrary, unknown digits. If $e+2d \leq r$ then the *RRNS* can correct the errors to reconstruct X [12].

Efficient error correcting algorithms are reported in [16-18], while an overview on *RRNS* is available in [19].

3 The Dependable and Secure Data Storage for Ad Hoc Networks (DS²)

In the Dependable and Secure Data storage for mobile ad hoc networks (DS²) [12] the mobiles cooperate by creating and sharing files. The system provides procedures to create, share and access the files. Once created a file can be written or removed only by its owner (the file creator).

Hereafter we assume that each mobile is assigned with a unique identifier ranging from 0 to $n-1$, and we will sometimes use the concise notation u_i to denote the mobile i .

The file creation procedure exploits an appropriate *RRNS* to encode a file. To this purpose u_i selects a set of $h+r$ moduli (pairwise prime positive integers) m_1, \dots, m_{h+r} with $m_p > m_{p-1}$ for each $p \in [2, h+r]$. The moduli are chosen among a set of available moduli computed offline. Since the maximum number which can be represented is limited by the range $M = m_{r+1} \cdot \dots \cdot m_{h+r}$ of the *RRNS*, files of sizes exceedingly the range are preliminary partitioned into records b_1, \dots, b_s of size b bits each, with $2^b \leq M$, and each record is encoded separately.

Record b_i is encoded in the *RRNS* of moduli m_1, \dots, m_{h+r} by the $(h+r)$ -tuple of residues $(x_{i,1}, \dots, x_{i,h+r})$. Each residue is sent to a different mobile currently reachable by the file creator, which in turn stores the residue in its storage. The assignment of mobiles to residues is arbitrary with the only constraint that different residues of the same record should be stored in different mobiles. Note that the mobiles storing the residue are not provided with any information about the modulo used to encode that residue.

The file owner maintains a file descriptor containing the set of the moduli used for encoding and s *record descriptors*. Each record descriptor contains the set of the list mobiles storing the residue digits of the record with the correct association between mobiles and residues. The file descriptor is kept secret by the owner.

Due to the encoding properties, the file records can be read separately. Record reading requires knowledge of correspondence between the mobiles storing the residue and the moduli used to encode the residues. Assuming that a mobile i owns a copy of the file descriptor, it can issue read requests to all the mobiles storing a given record b_i . During the read procedure some of the requested residues could be lost during the communication or even could be corrupted before reaching i . Once u_i receives a sufficient number of residues, it executes the decoding procedure based on the Chinese Remainder Theorem. If no residues are corrupted the decoding procedure returns the value of b_i , otherwise u_i will attempt to recover from the corrupted residues. In general, the original content of record b_i can be recovered only if the residues are decoded with the correct moduli and the multiplicity of the erasures e and of the errors d is such that $2d+e \leq r$.

File sharing is enabled by the file owner by distributing encrypted replicas of the file descriptor to trusted mobiles. In DS^2 it is assumed that mobiles sharing a file are not allowed to distribute the file descriptor to other mobiles, nor to write or remove the file. Since it is assumed that the mobiles sharing the file are trusted by the file owner, DS^2 does not employ any mechanism to inhibit distribution of descriptors. The policy of denying write and remove privileges is enforced by the mobiles hosting the residue digits. Failure to enforce this policy is equivalent to malicious digit corruption by the hosts.

The reader is referred to [12] for an extensive presentation of the DS^2 .

Table 1. Set of moduli used for the encoding

m_1	65536	m_{10}	65503	m_{19}	65449	m_{28}	65393	m_{37}	65339
m_2	65533	m_{11}	65501	m_{20}	65447	m_{29}	65383	m_{38}	65327
m_3	65531	m_{12}	65497	m_{21}	65437	m_{30}	65381	m_{39}	65323
m_4	65529	m_{13}	65491	m_{22}	65431	m_{31}	65371	m_{40}	65321
m_5	65527	m_{14}	65489	m_{23}	65423	m_{32}	65369	m_{41}	65311
m_6	65525	m_{15}	65479	m_{24}	65419	m_{33}	65363	m_{42}	65309
m_7	65521	m_{16}	65477	m_{25}	65413	m_{34}	65357	m_{43}	65293
m_8	65519	m_{17}	65473	m_{26}	65411	m_{35}	65353	m_{44}	65287
m_9	65509	m_{18}	65459	m_{27}	65407	m_{36}	65347	m_{45}	65281

4 Code Efficiency of DS²

We firstly evaluate the code efficiency without redundancy (that is, $r=0$). Let f be a file created by mobile i composed by s records b_1, \dots, b_s , where each record consists of b bits.

To improve the performance of the encoding/decoding procedures, we select the set of moduli such that most of the operations can be performed using single precision arithmetic. To this purpose we constructed a library of 45 moduli which is shown in Table 1. The largest modulo m_1 is 2^{16} , and all the other moduli have been chosen as close as possible to 2^{16} , with the constraint that the moduli must be pairwise prime.

Consider the Residue Number System with no redundancy (thus $r=0$) of the first h moduli of Table 1, and let $M = m_1 \cdot \dots \cdot m_h$ be its range and $b = \lfloor \log_2 M \rfloor$. Assume that record b_i ($t \in [1, s]$) is in the range $[0, 2^b)$ (it can be represented with b bits), and b_i is encoded into the set of residues $x_{t,1}, \dots, x_{t,h}$ where residue $x_{t,p}$ ($p \in [1, h]$) is encoded with 16 bits. Hence the entire record is encoded in $e=16h$ bits.

The code efficiency is defined as the ratio $\varphi=b/e$. The code efficiency of encoding with the first h moduli in the library of Table 1 is above 0.96 for $h \leq 6$, and it is above 0.99 for $h > 6$.

However, assuming range $[0, 2^b)$ for records, would imply that the length of the records, prior to residue encoding, would not be aligned to bytes. If this is a requirement, the length of records should be set to $b' = b - \beta'$, where β' is the smallest positive integer such that b' is a multiple of the byte length. For any choice of h in the library of Table 1, with $1 < h \leq 45$, $b' = 16h - 8$, and the record length b' scales linearly with h .

Under the latter assumption, Figure 1a depicts the code efficiency in terms of the ratio $\varphi = b'/e$ for different values of h . It is seen that, as h increases, the difference between b' and e remains constant while b' and e increase, and hence the code efficiency also increases.

We now evaluate the code efficiency using $r > 0$ redundant moduli. Let us consider the RRNS of the first $h+r$ moduli of Table 1. Since the redundant moduli should be larger than the non-redundant ones, the range is given by $M = m_1 \cdot \dots \cdot m_{h+r}$, and $b = \lfloor \log_2 M \rfloor$. Record b_i ($t \in [1, s]$) is encoded into the set of residues $x_{t,1}, \dots, x_{t,h+r}$, where

$x_{i,p}$ ($p \in [1, h+r]$) has length 16 bits, and the length of the encoded record is $e=16(h+r)$. Assuming that the length of the record, prior to residue encoding is aligned to the byte, and defining $b'=16h-8$. as above, the code efficiency $\varphi=b'/e$ has been evaluated for $r \in [1, 10]$ and $h \in [1, 35]$. As shown in Figure 1b the code efficiency increases rapidly for $h < 10$, after which it asymptotically approaches $h/(h+r)$.

5 Encoding/Decoding Complexity of DS²

Consider first the complexity of operations $(p+q)_m$ and $(pq)_m$, with $p, q \in [0, 2^l)$. For the sake of simplicity, we tolerate that the computation yields $[p+q]_m$ and $[pq]_m$ where $[x]_m$ denotes an integer in $[0, 2^l)$ congruent to $(x)_m$. This substitution is tolerable, and sometimes useful, in the application under consideration. It is assumed that $m=2^l-\delta$, with $l=16$ and $\delta < 2^8$. The latter assumption holds for all moduli in the library of Table 1.

Given integers $p, q \in [0, 2^l)$, let $p+q=a_1 2^l + b_1$ where a_1 and b_1 are non negative integers. From $p+q \leq 2^l + 2^l - 1$ and $m=2^l-\delta$ it is immediate that $a_1 \leq 1$, $b_1 \leq 2^l - 1$ and $[p+q]_m = [a_1 2^l + b_1]_m = [a_1 m + a_1 \delta + b_1]_m = [a_1 \delta + b_1]_m$. If $a_1=0$ then the single precision modulo is obtained. Else ($a_1=1$) let $\delta + b_1 = a_2 2^l + b_2$ with a_2 and b_2 non negative integers and $a_2 \leq 1$. Observe that $a_2=1$ implies $b_2 = \delta + b_1 - 2^l \leq \delta + 2^l - 1 - 2^l = \delta - 1$. Then $[\delta + b_1]_m = [a_2 2^l + b_2]_m = [a_2 m + a_2 \delta + b_2]_m = [a_2 \delta + b_2]_m$. From $b_2 \leq \delta - 1 < m$ follows that $[\delta + b_1]_m = \delta + b_2$ if $a_2=1$ and $[\delta + b_1]_m = b_2$ if $a_2=0$. Then the complexity of $[\delta + b_1]_m$ requires a single precision addition, and computing $[p+q]_m$ requires two single precision additions in the worst case.

Similarly, given integers $p, q \in [0, 2^l)$, let $pq=a_1 2^l + b_1$ where a_1 and b_1 are non negative integers. From $pq \leq 2^l(2^l-2)+1$, it is immediate that $a_1 \leq 2^l-2$, $b_1 \leq 2^l-1$, and $[pq]_m = [a_1 m + a_1 \delta + b_1]_m = [a_1 \delta + b_1]_m = [[a_1 \delta]_m + b_1]_m$. Letting $a_1 \delta = a_2 2^l + b_2$ with a_2 and b_2 non negative integers, from $a_1 \delta \leq (2^l-2)\delta \leq 2^l(\delta-1) + 2^l - \delta$, it follows $a_2 \leq \delta-1$. In turn, $[a_1 \delta]_m = [a_2 m + a_2 \delta + b_2]_m = [a_2 \delta + b_2]_m$. Since $a_2 \delta \leq (\delta-1)\delta < 2^l$ and $b_2 < 2^l$, computing $[a_1 \delta]_m$ requires 2 single precision additions in the worst case.

In conclusion, computing $[pq]_m$ requires at most 2 single precision multiplications (to yield the pairs (a_1, b_1) and (a_2, b_2)) and 4 single precision additions (2 additions to yield $[a_1 \delta]_m$ as $[a_2 \delta + b_2]_m$ and 2 additions to yield $[pq]_m$ as $[[a_1 \delta]_m + b_1]_m$).

Given an RRNS of the first $h+r$ moduli m_1, \dots, m_{h+r} of Table 1 and range $M=m_1 \dots m_{h+r}$, consider now the complexity of the procedure of residue encoding of an integer X in the range $[0, M)$. X is expanded as:

$$X = \sum_{i=0}^{h-1} (a_i 2^{il}) \quad (1)$$

The encoding procedure computes residues x_1, \dots, x_{h+r} , where residue $x_i = (X)_{m_i}$ for each $i \in [1, h+r]$. For the ease of notation, let $m=m_i$, $\delta=\delta_i$ for some $i \in [1, h+r]$. From (1) and $m=2^l-\delta$ follows that $x = \left(\sum_{i=0}^{h-1} (a_i (m + \delta)^i) \right)_m = \left(\sum_{i=0}^{h-1} (a_i \delta^i) \right)_m$, and then

$$x = \left(\sum_{i=0}^{h-1} (a_i (\delta^i)_m)_m \right)_m \quad (2)$$

The values of $(\delta^i)_m$ can be computed offline and stored with the moduli in the library of Table 1: this requires storing $4h$ single precision integers for each modulo. It is immediate from (2) that the computation of each residue requires at most h multiplications mod m and $h-1$ additions mod m , that is, $2h$ single precision integer multiplications and $4h+2h-2=6h-2$ single precision integer additions in the worst case.

In order to evaluate the complexity of decoding, assume without loss of generality that $h+t$ ($0 \leq t \leq r$) residues x_1, \dots, x_{h+t} from the encoding of a given record are received correctly, and let $M' = m_1 \dots m_{h+t}$. Then

$$X = \left(\sum_{i=1, h+t} \left(\frac{M'}{m_i} (x_i \beta_i)_{m_i} \right) \right)_{M'} \quad (3)$$

where $\beta_i = \left\langle \frac{M'}{m_i} \right\rangle_{m_i}$ ($i \in [1, h+t]$) is a single precision integer since $\beta_i < m_i$. From (3) the

decoding procedure involves the following operations:

1. $h+t$ multiple precision multiplications to yield $M'/m_i (x_i \beta_i)_{m_i}$;
2. $h+t-1$ multiple precision modular additions to yield the sum of the above products;
3. $h+t$ modular multiplications of single precision integers (that is, $2(h+t)$ single precision multiplications and $4(h+t)$ single precision integer additions) to yield $(x_i \beta_i)_{m_i}$ for each i ;
4. computation of β_i for each i .

Regarding 4), the multiplicative inverse β_i can be efficiently computed if the multiplicative inverse of MM_R/m_i is known as a constant for every i . Such single precision integer can be computed offline for each modulo in the library of Table 1.

It is easily seen that β_i is given by:

$$\beta_i = \left(\prod_{p=h+t+1}^{h+r} \left(\left\langle \frac{MM_R}{m_i} \right\rangle_{m_i} m_p \right)_{m_i} \right)_{m_i} \quad (4)$$

Evaluation of Equation 4 requires $r-t$ modular single precision multiplications, that is, in the worst case $2(r-t)$ single precision integer multiplications and $4t$ single precision, integer additions.

Considering that each multiple precision operation requires $O(h+t)$ operations, complexity of decoding $h+t$ residues has complexity $O((h+t)^2)$.

6 Security Issues in DS²

Confidentiality, authenticity and availability are among the classical security requirements. Confidentiality implies that only authorised users should be able to read a message; integrity implies that not authorized users are unable to modify a message, and the availability requirement consists in the protocol capacity to detect and resist to Denial of Service (DoS) attacks. In the following we analyse the compliance of DS² to such requirements.

Confidentiality. To recover a single record encoded into $h+r$ residues by DS² (where h is the number of non-redundant moduli and r is the number of redundant moduli), an attacker must know the nodes on which the residues of the record are stored and the correspondence between each residue and the appropriate modulo. Further, the residues of a record do not provide information about the record content. More specifically a record can be successfully read only if the multiplicity of the erasures e and of the errors d is such that $2d+e \leq r$ and the correspondence between available residues and the moduli is known. Under these assumptions the record content can be recovered using the Chinese Remainder Theorem. We are unaware of any efficient method to perform decoding which does not use the Chinese Remainder Theorem. However, the record can still be recovered by a brute force attack.

We evaluate the complexity of a brute force attack to decode the information of a record in the case in which $h+t$ correct residues $\{x_1, \dots, x_{h+t}\}$ are known. In principle an attacker may not know the values of h and r , however these indexes could be easily guessed by the attacker as the number of reasonable combinations is extremely limited, for this reason we assume that also h and r are known.

The attacker should consider all the possible $h+t$ -tuples of residues in association with all the permutations of the $h+r$ available moduli (which are public), and look for one combination leading to a legitimate number. Note that it is possible that several combinations lead to wrong legitimate numbers, but we disregard this possibility to the advantage of the attacker. It is then easy to see that the number of possible combinations is given by

$$\Theta = \binom{h+r}{h+t} (h+t)! = \frac{(h+r)!}{(r-t)!} \quad (5)$$

It should be considered however that the attacker has the additional advantage that the DS² encoding is error correcting. Hence for each combination it could execute the error correction procedure and look for the correct combination of at least $h + \lceil t/2 \rceil$ moduli and residues. Note that applying the error correcting procedure would increase the workaround factor required by the attacker to break the confidentiality, however in the following, we do not take into consideration such a factor. Hence, the resulting analysis is a lower bound on the efforts required to the attacker to break the confidentiality of the scheme for this type of attack. Note that, taking into account the possibility for the attacker to exploit the error correcting features of the RRNS, the number of combinations in Equation (5) should be divided by

$$\Phi = \sum_{k=0, \lfloor t/2 \rfloor} \left(\binom{h+t}{k} \frac{(r-t+k)!}{(r-t)!} \right), \text{ where, for each } k, \text{ the first factor accounts for}$$

all the possible combinations of k residues associated with the wrong moduli, and the second factor accounts for all the possible association of these residues with all the moduli which are not correctly assigned.

From Figure 2, which shows the value $C = \log_2 \frac{\Phi}{h!}$ for $r=8$, $t=4$, and $h \in [10,35]$ and the factorial of h , it is seen that the logarithm of the number of combinations C grows as the logarithm of $h!$.

As an alternative, the attacker may generate all the possible integers in the range $[0, M)$ and encode each integer using the set of available moduli, until it finds an integer X whose encoding produces a set of residues which includes the set $\{x_1, \dots, x_{h+t}\}$ of the received residues. It should be observed that this condition is not sufficient to guarantee that X equals the original encoded information, however we disregard this possibility to the advantage of the attacker. It is immediate that, disregarding the complexity of encoding, this procedure has complexity $O(M)$. This leads to a complexity of about $O(2^{16(h+r)})$, which however, considering the encoding with the moduli library of Table 1, results less efficient than the technique discussed above.

In principle a record could be reconstructed also by a mobile which has somehow received at least $t < h$ residue digits. Since the correspondence of residue digits with the moduli is unknown to the malicious mobile, it should then consider all possible t -tuples of residues in association with all permutations of the $h+r$ available moduli. Assuming that the correct association of a t -tuple of residues with the t moduli is somehow guessed, the decoding procedure only yield an integer which is congruent to the record. The record could be recovered by adding some (unknown) multiple of the product of some of the additional moduli.

Decoding the residues in this way leads to many legitimate numbers, and it may be very difficult to determine the right one. If the records are plain ASCII encoding, a clue may be provided by the fact that most legitimate numbers do not correspond to ASCII encoding, on the other hand, to save wireless bandwidth and mobiles storage, the encoding and dispersal of the file is most likely preceded by file compression and hence DS^2 generally operates on binary files.

Integrity. We consider any attempt of a malicious mobile to modify a record. Note that the resilience to this type of attack is independent whether the modification is meaningful or not, that is, the attacker tries to modify the record in such a way that the resulting record has a different, meaningful information content, or the attacker randomly modifies the record, which is (wrongly) recognised as a correct one by the legitimate requester, despite its information content.

With the DS^2 encoding, malicious corruptions of residues can be recovered if $h+t$ ($t \leq r$) residues can be read and no more than $\lfloor t/2 \rfloor$ residues are corrupted, and can be detected if the number of corrupted residues does not exceed t . This feature of DS^2 is an improvement with respect to the behaviour of recent standard protocols, which are subject to this type of threat, as in [21].

Availability. For every read operation, the disruption of up to r residues of a record do not compromise the capability recovering the record. Moreover, if the residues are

routed through paths which have minimal intersection points (the ideal situation would be to have disjoint paths), the probability of DS^2 to be resilient to a DoS attack enhances dramatically. Indeed, for the attacker to be successful, it should take control of at least $r+1$ different specific nodes, each one being on one of the disjoint paths.

As a final remark, we note there is a number of little security flaws to which DS^2 is exposed, but which can be easily circumvented. For example, if the encoding produces a residue containing the value 65535, it is immediate that this residue must correspond to modulo $m_0=65536$. Also, the encoding of a record whose decimal content is smaller than any module of the $RRNS$ yields residues whose content is the same as the original record. These little flaws can be easily prevented, for example by selecting a random constant (to be kept secretly in the file descriptor) to be added modulo 65536 to each residue after the encoding.

7 Comparison of DS^2 with Information Dispersal Algorithm (IDA)

We now compare DS^2 with the Rabin's Information Dispersal Algorithm (IDA) [10]. To this purpose we briefly review IDA.

Let us consider a file f composed by N characters b_1, \dots, b_N where each b_i is in the range $[0, B)$, and let p be a prime with $p > B$. Typical values of B and p are $B=256$ and $p=257$. As in DS^2 the file encoding produces $h+r$ fragments to be dispersed in the network, and the file content can be reconstructed from any h fragments.

Let us assume for the sake of simplicity that $N=(h+r)h$. To the purpose of encoding, IDA partitions the file in $h+r$ records S_1, \dots, S_{h+r} , each of h characters, that is, $S_i=(b_{h(i-1)+1}, \dots, b_{ih})$ for each $i \in [1, h+r]$. Then it selects a $h+r$ -tuple (a_1, \dots, a_{h+r}) of $h+r$ randomly chosen vectors such that $a_i=(a_{i1}, \dots, a_{ih})$, $a_{ij} \in [0, p)$, and any subset of h vectors is linearly independent.

The encoding produces $h+r$ fragments F_1, \dots, F_{h+r} each of size $h+r$ characters, where $F_i=(c_{i1}, \dots, c_{ih+r})$ and $c_{ik}=(a_i \times S_k) \bmod p$ for each $i, k \in [1, h+r]$.

Note that the characters of the fragments F_1, \dots, F_{h+r} are in the range $[0, p)$, while the characters of the file are in the range $[0, B)$ with $B < p$. This in general leads to a wastage of at least 1 bit per character, however with a simple technique [10] this overhead can be avoided.

The decoding procedure of IDA requires at least h fragments of the encoded file. Assume that F_1, \dots, F_h are available, the file can be reconstructed as follows. Let \mathbf{A} be the $h \times h$ matrix whose i th row is a_i . Then record S_k is given by:

$$S_k = \mathbf{A}^{-1} \begin{bmatrix} c_{11} \\ \dots \\ c_{h1} \end{bmatrix} \quad (6)$$

Disregarding the memory overhead to store the vectors, the IDA's encoding is optimal, since the overhead for an r -erasure tolerance encoding of a file is $(h+r)/h$. Let-

ting a record size of $b = \lfloor \log_2 M \rfloor$ in DS^2 , the code efficiency of IDA and DS^2 is almost the same.

Encoding/Decoding Efficiency. It is immediate that for IDA, the time complexity of the encoding of a file of size $N=(h+r)h$ is $O(N(h+r))$ plus the cost of generating the set of vectors, while the complexity of decoding is $O(hN)$ plus the cost of inverting matrix \mathbf{A} . Although for sufficiently large files the costs involved by the generations of the vectors and the matrix inverse becomes negligible, Rabin shown efficient techniques for generating matrices which are invertible in time $O(h^2)$.

Considering a file of size $8N$ bits, with $N=(h+r)h$, the complexity of encoding the file with IDA is $O(h(h+r)^2)$. In the encoding of the same file with DS^2 , the file is preliminary partitioned into $8N/b$ records with $b \approx 16h$. Hence the encoding with DS^2 has complexity $O(h(h+r)N/h) = O(h(h+r)^2)$, which equals the complexity of the encoding of IDA. The complexity of the file decoding with IDA is $O(h^2(h+r))$, while the decoding with DS^2 is, in general, $O((h+r)^3)$ which is slightly worse than the decoding complexity of IDA. However, if only erasures are considered, the decoding with DS^2 is $O(h^2(h+r))$, which equals the complexity of decoding with IDA. Table 2 summarizes the different encoding/decoding running time for the two algorithms.

Table 2. Encoding/Decoding Efficiency.

	DS^2	IDA
Encoding	$O(h(h+r)^2)$	$O(h(h+r)^2)$
Decoding $h+r$ residues	$O((h+r)^3)$	-
Decoding (only erasures)	$O(h^2(h+r))$	$O(h^2(h+r))$

Confidentiality. In IDA, the recovery of a file requires knowledge of (a) the vectors of the matrix \mathbf{A} ; (b) the relative positions of the vectors of matrix \mathbf{A} ; (b) the knowledge of the nodes where the fragments are located; (c) the knowledge of the correspondence of each fragment with the appropriate position in the vector to be multiplied for \mathbf{A}^{-1} , as in Equation (6). It is straightforward to map the requirements for IDA and those for DS^2 . Indeed, in DS^2 what is needed is (a) the knowledge of the values of the moduli; (b) the knowledge of the nodes where the residues are located; (c) the knowledge of the correspondence of each residue with the appropriate module.

If the set of moduli is kept secret in DS^2 and the vectors composing matrix \mathbf{A} is kept secret in IDA, the file sharing requires to transmit to the intended receivers such data. In this case the overhead of transmission of the $(h+r)$ moduli requires at most $16(h+r)$ bits, which is lower than the overhead required to transmit the matrix \mathbf{A} (which accounts for $h(h+r)\log_2 p$ bits). However, in the following we will assume that the set of moduli is publicly known, as well as the vectors of the matrix \mathbf{A} (but not their relative position). Under this hypothesis assume that that $h+t$ fragments are correctly recovered. Then, the security of IDA relies on the workload required to the attacker to correctly identify the correct position of vectors to obtain matrix \mathbf{A} , and bound the fragment with the appropriate position of the vector to be multiplied for \mathbf{A}^{-1} , as in Equation (6). The security analysis for these points results in Equation (5).

However, in IDA also the access to less than h fragments may provide some information about the file. For this reason, to enforce confidentiality, the file could be encrypted before being encoded and dispersed by IDA, thus introducing additional overhead.

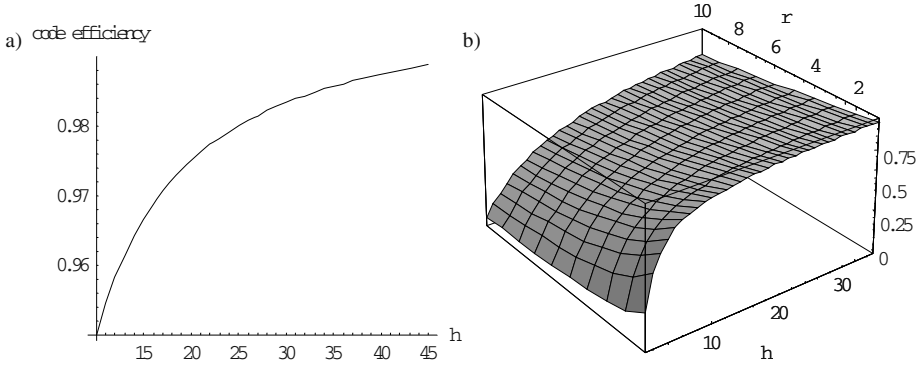


Fig. 1. Code efficiency for: (a) $h \in [10,45]$ and record size $b' = 16h - 8$, and (b) $r \in [1,10]$, $h \in [1,35]$, and record size $b' = 16h - 8$.

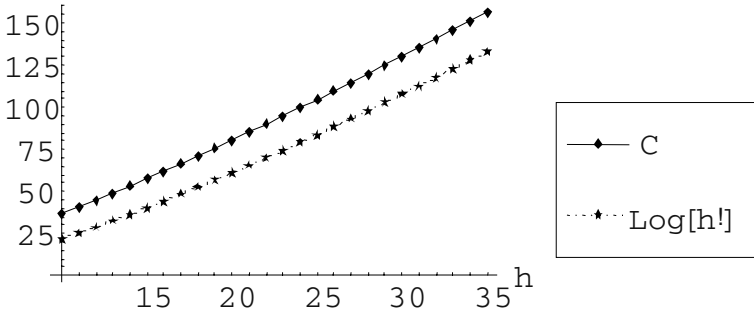


Fig. 2. Number of combinations C to be analyzed in case of brute force attack compared with $h!$ ($r=8$, $t=4$, and $h \in [10,35]$).

On the other hand, in DS^2 no information can be easily recovered from any subset of residues if the association between moduli and residues is not known, and a brute force attack would require a number of operations which grows as the factorial of h (for values of $h \leq 50$).

Availability and Integrity. DS^2 employs an *RRNS*-based encoding which features error correcting capabilities, such that corruptions of less than $r/2$ residues can be recovered, and corruptions of up to $r/2$ residues can always be detected. Hence the availability and integrity requirements are easily met in DS^2 .

As compared to DS^2 , the IDA encoding is designed to cover only erasures. This means that, in case of malicious corruption of fragments, the original content of the

file cannot be recovered. A solution could be to introduce fingerprints, that is, an indicator of whether the fragment has been altered or not, and hence discarded in the former case. However, the use of fingerprints introduces both computational and storage overhead to the encoding. Resorting to standard fingerprinting functions, like SHA-1, MD5 [20] has a computational complexity which is linear in the size of the block, but requires at least 128 bits per fragment. Note that without fingerprinting, both availability and integrity are at a stake. Moreover, only fingerprinting is not enough to prevent violation of the integrity requirement. Indeed, an attacker could replace the fragment F_i with an arbitrary one (say F'_i), compute the fingerprinting ($H(F'_i)$) for this fragment and then send the tuple $(F'_i, H(F'_i))$ which will pass the integrity check. This would result in violation of the security requirements. This problem can be solved using keyed-fingerprinting [20], that is, the fingerprint is a function of two parameters, a key (shared only by legitimate users) and the message. Hence, the tuple $(F'_i, H(k, F'_i))$ can not be replaced, unless key k is known.

Both integrity and availability of IDA can be enforced resorting to keyed fingerprinting. However, this would result in an additional overhead in both computation and storage. While the computational overhead can be considered negligible, the storage requirement is not. Furthermore, the longer the fragment is, the more it is subject to possible errors during the transmission. Once a corrupted fragment reaches destination, it would not pass the integrity check, and hence will be discarded. Hence, to achieve the same degree of reliability and security as DS^2 , IDA might require an higher degree of redundancy.

8 Conclusions

This paper discusses a dependable and secure data storage for mobile, wireless networks (DS^2) based on a peer-to-peer paradigm. DS^2 provides support to create and share files under a write-once model, and ensures at the same time data confidentiality and dependability by encoding files in a Redundant Residue Number System. More specifically files are partitioned into records and each record is encoded separately as $(h+r)$ -tuples of data residues using $h+r$ moduli. In turn, the residues are distributed among the mobiles in the network. Dependability is ensured since data can be reconstructed in the presence of up to $s \leq r$ residue erasures, combined with up to $\lfloor \frac{r-s}{2} \rfloor$ corrupted residues, and data confidentiality is ensured since recovering the original information requires knowledge of the correspondence between moduli and residues. The achievable degrees of dependability and security are determined by the choice of the *RRNS* (that is, of the set of moduli).

The paper analyzes the code efficiency of DS^2 using a library of moduli which allows for efficient encoding and decoding procedures based on single precision arithmetic, and discusses the security issues. The comparison of DS^2 with IDA shows that the two approaches have almost the same performance in terms of code efficiency and complexity, even though DS^2 provides richer security features than IDA, ex-

plotting the *RRNS* codes. To achieve the same level of security of DS^2 , IDA requires additional overhead in both computation and storage resources.

References

1. D. B. Johnson and D. A. Maltz: Dynamic source routing in ad hoc wireless networks. Mobile Computing, edited by T. Imielinski and H. Korth, chapter 5, pp. 153–181. Kluwer Academic Publishers (1996)
2. C. E. Perkins and E. M. Royer: Ad-hoc on-demand distance vector routing. Proc. IEEE 2nd Workshop on Mobile Computing Systems and Applications, New Orleans, LA (1999) 90–100
3. L.M. Feeney and M. Nilsson : Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. Proc. IEEE 20th INFOCOM, Vol. 3, Anchorage AK (2001) 1548 –1557
4. C. A. Thekkath, T. Mann, and E. K. Lee : Frangipani: A Scalable Distributed File System. Proc. ACM 17th Symposium on Operating Systems Principles (1997) 224–237
5. Satyanarayanan, M.; Kistler, J.J.; Kumar, P.; Okasaki, M.E.; Siegel, E.H.; Steere : Coda: a highly available file system for a distributed workstation environment. IEEE Transactions on Computers, Vol.39 (4) (1990) 447–459
6. T. E. Anderson, M. Dahlin, J. Neefe, D. Patterson, D. Roselli, and R. Wang: Serverless Network File System. Proc. ACM 15th Symposium on Operating System Principles, Copper Mountain Resort, Colorado (1995) 109–126
7. J. C. Fabre, Y. Deswarte, and B. Randell. “Designing secure and reliable applications using fragmentation-redundancy-scattering: an object-oriented approach. Proc. 1st European Dependable Computing Conference, Berlin, Germany (1994) 21–38
8. Y. Chen, J. Edler, A. Goldberg, A. Gottlieb, S. Sobti, and P. Yianilos: A Prototype Implementation of Archival Intermemory. Proc. ACM 4th Conference on Digital libraries, Berkeley, CA (1999) 28–37
9. J. Kubiawicz et. Al.: OceanStore: An Architecture for Global-Scale Persistent Storage. Proc. ACM 9th Conference on Architectural Support for Programming Languages and Operating Systems, Cambridge, MA (2000) 190–201
10. M. O. Rabin : Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance. Journal of the ACM, Vol.36, (2) (1989) 335–348
11. L. Rizzo : Effective Erasure Codes for Reliable Computer Communication Protocols. ACM Computer Communication Review, Vol. 27 (2) (1997) 24–36
12. S. Chessa and P. Maestrini : Dependable and Secure Data Storage and Retrieval in Mobile, Wireless Networks. Proc. IEEE DSN 2003, International Conference on Dependable System and Networks, San Francisco, USA (2003)
13. F. Barsi and P. Maestrini : Error Correcting Properties of Redundant Residue Number Systems. IEEE Transactions on Computers, Vol. C-22 (3) (1973) 307–315
14. D. Mandelbaum : Error Correction in Residue Arithmetic. IEEE Transactions on Computers, Vol. C-21 (6) (1972) 538–545
15. N. S. Szabo and R. I. Tanaka, Residue Arithmetic and its Applications to Computer Technology, Mc Graw-Hill, New York (1967)
16. D. Mandelbaum : On a Class of Arithmetic Codes and Decoding Algorithm. IEEE Transactions on Information Theory, Vol. IT-21 (1976) 85–88

17. F. Barsi and P. Maestrini: Improved Decoding Algorithms for Arithmetic Residues Codes. *IEEE Transactions on Information Theory*, Vol. IT-24 (1978) 640–643
18. J. D. Sun and H. Krishna: A coding theory approach to error control in redundant residue number systems. II. Multiple Error detection and correction. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 39 (1) (1992) 18–34
19. M. Soderstrand, W. K. Jenkins, G. A. Jullien and F. J. Taylor, *Residue Number Arithmetic: Modern Applications in Digital Signal Processing*, IEEE Press, NY (1986)
20. M. Bellare, R. Canetti, and H. Krawczyk: Keying hash functions for message authentication. in *Advances in Cryptology – Crypto 96 Proceedings*, LNCS Vol. 1109, N. Koblitz ed, Springer-Verlag (1996)
21. Nikita Borisov, Ian Goldberg, David Wagner: Intercepting mobile communications: the insecurity of 802.11. *MOBICOM 2001*, Rome, Italy (2001) 180–189

A Comparative Analysis on Performance of Mobile IP with Paging Support

Hung Tuan Do and Yoshikuni Onozato

Department of Computer Science, Gunma University, Kiryu, Gunma 376-8515, Japan
{hung, onozato}@nzt1.cs.gunma-u.ac.jp

Abstract. This paper presents a comparative analysis on the signaling cost functions of Mobile IP (MIP) with different paging protocols and schemes and investigates constructing optimal paging areas using discrete system model as a mobility model. In wireless mobile Internet, mobile users often visit foreign networks that might be far away from their home networks and the occurrences of their inter-domain movement are relatively rare. In this scenario, our analytical results show that paging, particularly individual paging, can significantly improve the total signaling cost of MIP. We show that Domain paging can bring about considerable cost saving compared to FA (Foreign Agent) paging. Our results also demonstrate the significant advantages of Individual paging over Aggregate paging. The results show that specifying the optimal PA size is critical in saving signaling cost of MIP with paging support.

1 Introduction

At present, MIP is the current standard protocol for the mobility management at IP layer. Unfortunately, Mobile IP (MIP) was first designed without consideration of the performance efficiency and QoS warranty. We believe that, in addition to the popular approach of localization of location updates, paging is an important approach to improve the performance of MIP [1], [2], [3].

In wireless mobile networks, paging is a process to determine the exact location of a specific Mobile Terminal (MT) in PCSs or a Mobile Host (MH) in Mobile IP that is in stand-by state. Paging service is popularly deployed in wireless WAN for two major benefits: to reduce location update cost and to save power consumption of mobile terminals.

With paging support, an idle MH performs location update less frequently (at each change of paging area) than it does in the base MIP (at each change of subnet). Thus, the application of paging service introduces two main benefits to MIP: reduce location update cost and save power consumption of MHs. Adversely, paging procedure also generates additional signaling overhead that is paging cost itself and the latency in locating an MH for packet delivery. There exists a tradeoff between paging cost and registration cost, namely with bigger paging area (PA) size, an MH tends to update its location less frequently, thus the location update cost is reduced, but the cost of paging over the PA certainly increases and vice versa. Therefore, it is desirable to figure out the optimal paging area size that minimizes the total signaling cost.

In literature, there is a couple of research works addressing the problem of MIP with paging [1], [2], [3]. In Ref. [3], the authors propose P-MIP (Paging Extensions for MIP). In our taxonomy that is presented in section 2, the paging protocol and paging scheme of P-MIP are very similar to FA paging protocol and Aggregate paging scheme, respectively. The performance of P-MIP is investigated using a combination of analysis and simulation. In analysis, the authors present signaling costs of MIP and P-MIP with the assumptions that paging areas and wireless cells are square-shaped and that user mobility model is Fluid Flow - an aggregate mobility model. The paper then compares the signaling costs of MIP and P-MIP under the effects of the numbers of cells in a paging area. The analytical results show that paging could improve the performance of MIP by reducing the signaling cost in well-designed ranges of parameters such as paging area size and active timer.

In [2], the author analyzes the performance of MIP with the proposed paging schemes by simulation using a modified Random Walk as the user mobility model. In this work, paging areas are assumed to be hexagonal and cellular network is assumed as uniform grid of hexagonal cells. Various paging schemes are considered here, namely *mcpf* (paging with fixed paging area size), *mcp*, *mcp1*, and *mcp_opt* (all are individual paging schemes but with different decision algorithms of paging area update). It is shown that paging can bring about cost gain to MIP and that among paging schemes, individual paging schemes have advantages over *mcpf*, and among individual paging schemes, *mcp_opt* can outperform the other schemes. The performance of Hierarchical MIP (HMIP) with paging is also investigated. The combination of *mcp_opt* paging scheme with HMIP is shown to be the best solution among those considered. It should be noted that, the paging protocol in [2] is FA paging protocol and the individual paging schemes are different from ones in our analysis in PA construction and PA update algorithm.

In [1], the authors first proposed complete paging protocols for MIP, namely Home Agent (HA) paging, Foreign Agent (FA) paging, and Domain Paging. These three paging protocols for MIP are evaluated using simulation with three paging algorithms, namely Fixed Paging, Hierarchical Paging, and Last-Location Paging. In our classification, these paging algorithms belong to aggregate paging, i.e. the paper does not address individual paging schemes. In the simulation, no mobility model is assumed in the simulation, but mobility and call traces are employed instead. Regarding to HA updates and paging latency, the paging protocols and paging algorithms are compared. It is demonstrated that Domain paging performs best and HA paging performs worst among three proposed protocols.

Regarding performance evaluation (by analysis or simulation), some shortcomings of them can be seen. First, none of them compares the performance of MIP with different paging protocols and schemes by analysis. Second, the mobility models (e.g. Random Walk [2], Fluid Flow [3]) and assumptions have severe limitations: subnets are assumed to be squared [3], [1] or hexagonal [2]; movement history is not considered. In the Internet, a subnet takes no specific shape and the distance between two end points should be measured by number of hops between them [4]. Thus, any assumption on the geometric shape of a subnet might be far from realistic. Furthermore, the location (state) of a user depends not only on his current location (state), but also on his movement history (past states). The Fluid Flow model [2] is not a good mobility model for pedestrians or mobile users moving with different velocities. In [1], the

performance of paging protocols is measured by the number of HA location updates and paging latency. We propose that a more complete signaling cost function be employed to evaluate the performance of MIP with paging support.

A comparative study by simulation of Mobile IPv6 and IP paging for dormant mode location update in the scenarios of macro-cellular and hotspot networks is presented in Ref. [6]. From the initial simulation results for the case of fixed paging area size, the authors concluded that Mobile IPv6 dormant mode location management is comparable to IP paging, and hence, IP paging is unnecessary. However, there is difference in terms of configuration assumptions between this paper and our work. In this paper, a subnet consists of a group of cells and an MH needs to update its location at each subnet change, not at each cell change. In our work and Refs. [1], [3], each wireless cell exists in its own subnet, and in the base MIP, an MH is required to update its location per each inter-subnet (inter-cell) movement. In the sense of location update mechanism of dormant MHs, each subnet as a group of cells in this paper acts in the same way as a PA in our work. Furthermore, macro-cell in the simulation is assumed to be square-shaped, but then, the shape of subnets as a group of such macro-cells is not always the case in a spatial Internet scenario. As shown later in our analysis, total signaling cost of MIP with paging (or IP paging) is very sensitive to PA size, and MIP with paging is not always better than the base MIP with an arbitrary PA size. Therefore, PA size is a design parameter, and calculating the optimal PA size is critical in order to minimize the signaling cost of MIP with paging support. Unfortunately, the simulation in this paper is conducted only with fixed PA sizes (4 subnets and 9 subnets).

Moreover, an effort to enhance the performance of MIP is Route Optimization Protocol (IETF RO) [7], which is introduced to solve the problem of triangular routing. Route Optimization is already integrated into MIPv6. In Ref. [8], the authors proposed a method using Markovian Decision model to solve optimization problem of the total cost as the sum of link cost and signaling cost in order to improve further the performance of IETF RO.

The contribution of this paper is as follows. We evaluate the performance of MIP with different paging protocols and paging schemes comparatively by analysis. We address two major limitations above by using Discrete System model [4] as the user mobility model with two distinct features: arbitrary subnet topology and consideration of user movement history. Based on the mobility model and the analysis of the operational models of paging protocols and schemes, we derive the total signaling cost functions of MIP with different paging protocols and schemes. We then analyze the cost functions of MIP with different paging protocols and paging schemes under the impact of varying parameters: PA size, incoming call rate and user dwell time. Also, we determine the optimal PA size by using an iterative algorithm and show that specifying the optimal PA is crucial in saving the signaling cost of MIP with paging.

The remaining sections of this paper are organized as follows. In Section 2, we describe and analyze paging protocols and paging schemes. Section 3 formulates the signaling cost functions of MIP with paging protocols and paging schemes. Our numerical results are presented in Section 4. Finally, Section 5 concludes the paper and mentions some directions of our future works.

2 Paging Protocols and Paging Schemes

In this paper, we differentiate between the two terms: paging protocols and paging schemes. Paging protocol, as defined in [1], determines the node that initiates paging process and messages exchanged among nodes. A complete discussion of paging protocol can be found in [1]. The term “paging scheme” in this paper, on the other hand, determines how paging area and its size are specified.

2.1 Paging Protocols

In this analysis, we investigate two protocols: Domain paging and Foreign Agent (FA) paging [1]. For Home Agent (HA) paging [1], HA is the very paging initiator. However, HA is also a potential point of failure and bottleneck in MIP. Moreover, for an MH, the HA might be far away from the foreign network in which the MH is visiting. Therefore, we argue that initiating a paging process from the HA is not a practical approach. One more thing should be noted is that here we consider MIP in the environment of spatial wireless Internet, not MIP in cellular architecture.

2.1.1 FA Paging Protocol

The message flows of MIP with FA paging protocol [1] are shown in Fig. 1. The first packet destined for the MH is tunneled from the HA to the last-registered FA (FA1). Subsequently, FA1 buffers the packet and sends paging request to the other FAs in the PA then these FAs page over the air (message 1).

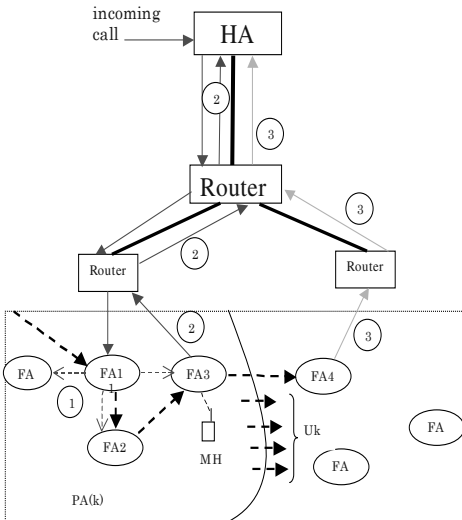


Fig. 1. Message Flows of MIP with FA Paging Protocol

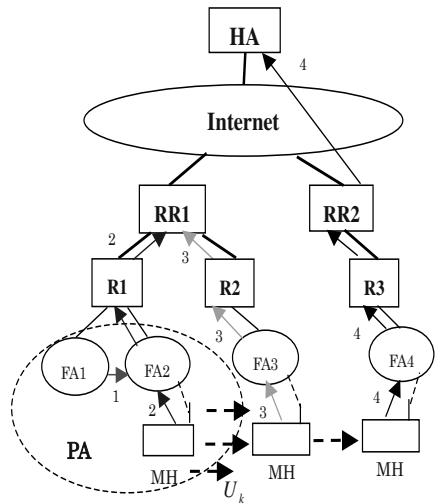


Fig. 2. Message Flows of MIP with Domain Paging Protocol

The MH registers its new FA with the HA via the location update message following a paging (message 2), which needs to be acknowledged by the HA, and at the same time informs the previous FA. The previous FA forwards the buffered packet to the MH and subsequently the HA delivers packets to the MH via its new FA. Message 3 indicates the location update binding when the MH moves out of the paging area with the rate U_k . Also this message needs to be acknowledged by the HA.

2.1.2 Domain Paging Protocol

Domain paging [1] is a router-assisted paging scheme as shown in Figure 2. The domain root router (RR) is the gateway into each domain. When a MH in standby state moves out of the current paging area with the rate U_k but still in the domain, it sends a location update message only to the domain RR (RR1) instead of its HA (message 3), which might be far away (intra-domain mobility). The message is sent hop-by-hop from the MH's FA to the domain root router, thereby creating new routing and paging state on each node (router/BS) in the path.

When an incoming packet destined for the MH in standby mode, a router or FA along the path from the domain root router to the last-registered FA of the MH selects itself to be the page initiator. The page initiator (FA1) then buffers and sends out a page request to all the FAs in the PA and these FAs page over the air (message 1). The MH replies to the page by sending a page response to the page initiator hop-by-hop, thereby updating its location and the route for future packet delivery.

For simplicity, we consider the case where the last-registered FA (FA1) is the page initiator and the page response is always sent to the domain root router to update the routing path to the MH (message 2). Actually, this can be seen as the case of HMIP with FA paging. Only if the MH moves out to another domain (domain 2) then it needs to send a location update binding to its HA (message 4). This is known as inter-domain mobility. In fact, the occurrences of inter-domain mobility in the Internet are relatively rare [1] and will be ignored in this paper.

The major difference and also advantage of HMIP with FA paging over MIP with FA paging is the localized location registration of MHs. Moreover, domain paging helps improve significantly the reliability of the system owing to its distributed manner in selecting page initiators.

2.2 Paging Schemes

The following paging schemes are considered in this paper:

1. *Static Aggregate Paging (SAP)*: the paging area is designed and fixed by the network administrator for all mobile users.
2. *Individual Paging*: each user specifies his own paging area (PA) according to his mobility and call pattern, i.e., if the PA size is k (in terms of the number of subnets covered) then PA will consist of the first k different visited subnets, similar to the idea of location area construction presented in [4]. In this category, we classify Individual paging further as follows:
3. *Static Individual paging (SIP) scheme*: The optimal PA size (in terms of the number of subnets covered) k_{opt} is pre-computed before communications and will not change during the communications.

4. *Dynamic Individual paging (DIP) scheme*: k_{opt} is adaptive to the user's current mobility and call parameters.

3 Total Signaling Cost

3.1 Assumption and Parameters

All parameters and their notations used in our analysis are shown in Table 1.

Table 1. List of parameters

Parameters	Meanings	Units
k	Size of paging area	subnets
λ	Incoming call/packet rate	calls/minute
U_k	Location update rate per MH due to its movement	updates/minute
C_p	Paging cost in a subnet per call	bytes/paging/subnet
C_u	Location update cost per hop	bytes/update/hop
d_{mh}	MH-HA distance (number of hops)	hops
d_{mg}	MH-GW distance (number of hops)	hops
d_{fp}	New FA – last-registered FA distance	hops
T_d	Dwell time (residence time) of MH in a subnet	minutes
α	Wired communication cost weight	
β	Cost weight ratio of wireless to wired communication	

In this paper, we make the following assumptions:

1. The MH-HA distance d_{mh} and MH-GW distance d_{mg} are assumed as fixed during the time of investigation.
2. Successive calls are not overlapped, i.e. the MH always accomplishes a call then turns into standby mode before the next call arrives. With this assumption, the MH is paged at each incoming call.
3. The last-registered FA is the paging initiator in Domain Paging and the MH always send its page response to the domain RR instead of the paging initiator. We also assume that all N subnets considered in our mobility model are located within a domain; and inter-domain mobility is so rare [1] that it can be ignored in our analysis.

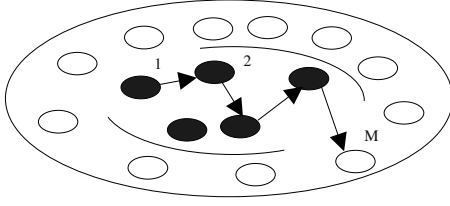


Fig. 3. SAP scheme model

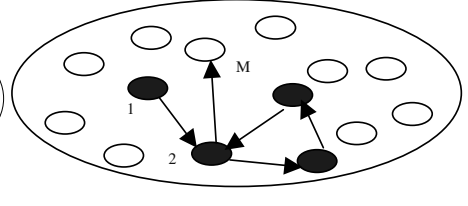


Fig. 4. Individual Paging scheme model

3.2 PA-Boundary-Crossing Rates

Our mobility model is adapted from the Discrete System model [4]. We assume that each MH can move randomly among N subnets and the PA consists of k subnets. The MH moves from one subnet to any of $(N-1)$ others with the equal probability $1/(N-1)$. The action an MH moves out of a subnet is called “a movement”. Let M be a random variable so that the MH moves out of the PA at movement M [4]. This model well captures the spatial nature of the Internet: a subnet can take an arbitrary shape.

In SAP scheme, the PA and its size are pre-defined and the same for all MHs. Similar to [4], the expectation of M in SAP can be derived:

$$E[M_{SAP}] = 1 + \frac{N-1}{N-k} . \quad (1)$$

Now, the PA-boundary-crossing rate U_k , which is exactly the HA update rate due to the movement of the MH, can be obtained (where T_d is the average dwell time):

$$U_k = \frac{1}{E[M_{SAP}]T_d} . \quad (2)$$

Contrary to SAP, in Individual Paging, PA and its size are specified adaptively for each MH. PA will comprise the first different k subnets visited by the MH, given PA size k . Similar to [4], the expectations of M in these schemes are:

$$E[M_{SIP}] = E[M_{DIP}] = 1 + (N-1) \sum_{i=1}^k \frac{1}{N-i} . \quad (3)$$

The PA-boundary-crossing rate U_k in this scheme can be obtained:

$$U_k = \frac{1}{E[M_{SIP}]T_d} = \frac{1}{E[M_{DIP}]T_d} . \quad (4)$$

3.3 Derivation of Signaling Cost Functions

3.3.1 MIP with FA Paging Protocol

In this analysis, for the sake of comparison, it is not necessary to consider the cost of packet delivery as it is similar to the process in MIP. From our analysis of message flows in MIP with FA paging protocol above, the total signaling cost function comprises of three elements: *paging cost* P (message 1), *the cost of location update following each page* U_1 (message 2) and *the cost of location update due to the movement of the MH* U_2 (message 3).

a. Paging Cost

Paging cost consists of the cost of paging via air interfaces of all k subnets in the paging area and the cost of wired communication when the last-registered FA contacts other FAs in the paging area via wired links and routers. Let N_r be the number of routers via that the paging initiating FA needs to send packets to other FAs. The paging cost is thus:

$$P = \lambda[kC_p + (k-1)\alpha C_u + N_r\alpha C_u] . \quad (5)$$

For simplicity, we assume that all FAs in current paging area are connected via one router, e.g. when there is only one level of hierarchy of FAs. The paging initiating FA contacts the router then the router will contact $(k-1)$ other FAs in the paging area. Consequently, the paging cost is simplified as follows:

$$P = k\lambda(C_p + \alpha C_u) . \quad (6)$$

b. Cost of Location Update Following Each Page

The registration of the MH's new FA to the HA with acknowledgment requires the cost of $2\alpha C_u(d_{mh} + \beta)$. Moreover, the communication cost when the new FA contacts the last FA and the last FA forwards the buffered packet to the new FA is $2d_{fp}\alpha C_u$. Totally, the cost of location update following an incoming call can be obtained as:

$$U_1 = 2\lambda\alpha C_u(d_{mh} + d_{fp} + \beta) . \quad (7)$$

With the assumption of one level hierarchy of FAs, we have $d_{fp} = 2$. More generally, we can assume $d_{fp} = \sqrt{k}$ [3] where k is the number of subnets in the paging area.

c. *Cost of Location Update Due to the Movement of the MH*

Whenever an MH moves out of its current paging area, it is required to register its new location (FA) with its HA. We denote U_{SAP_2} and U_{IP_2} as the location update costs of SAP and Individual paging, respectively. From (1), (2), (3) we can obtain:

$$U_{SAP_2} = 2\alpha C_u (d_{mh} + \beta) \frac{1}{E[M_{SAP}]T_d} = 2\alpha C_u (d_{mh} + \beta) \frac{N-k}{(2N-k-1)T_d} . \quad (8)$$

$$U_{IP_2} = 2\alpha C_u (d_{mh} + \beta) \frac{1}{E[M_{IP}]T_d} = 2\alpha C_u (d_{mh} + \beta) \frac{1}{[1 + (N-1) \sum_{i=1}^k \frac{1}{N-i}]T_d} . \quad (9)$$

d. *Total Cost Function*

The total signaling cost functions of MIP with FA paging protocol and SAP, SIP and DIP are denoted S_{SAP} , S_{SIP} and S_{DIP} , respectively. For each scheme, the total signaling cost is the summation of three elements described above. Accordingly, we have:

$$S_{SAP}(k, \lambda, T_d) = P + U_1 + U_{SAP_2} . \quad (10)$$

$$S_{SIP}(k, \lambda, T_d) = S_{DIP}(k, \lambda, T_d) = P + U_1 + U_{IP_2} . \quad (11)$$

3.3.2 MIP with Domain Paging Protocol and Hierarchical MIP with FA Paging Protocol

As we argued before, MIP with Domain paging in this scenario is similar to HMIP with FA Paging in the facet of signaling cost, and thus the analytic results here are applicable to both of them.

Similar to MIP with FA paging, the signaling cost of MIP with Domain paging comprises of three elements (Fig 2): *paging cost* P (message 1), *the cost of location update following each page* U_{D_1} (message 2) and *the cost of location update due to the movement of the MH* U_2 (message 3).

a. *Paging Cost*

This cost element in MIP with Domain paging is exactly similar to that of MIP with FA paging as in equation (5).

b. *Cost of Location Update Following Each Page*

In MIP with Domain paging, following a page, the MH sends its page response message to the domain RR instead of the HA, hence, similar to (6), we have:

$$U_{D_{-1}} = 2\lambda\alpha C_u (d_{mg} + d_{fp} + \beta) . \quad (12)$$

c. *Cost of location update due to the movement of the MH*

We denote $U_{SAP_D_2}$ and $U_{IP_D_2}$ as the costs of location update due to the movement of the MH in SAP scheme and Individual paging scheme, respectively. When the MH moves out of its current PA, it makes a location binding update to the domain RR, so similar to (7) and (8), $U_{SAP_D_2}$ and $U_{IP_D_2}$ are as follows:

$$U_{SAP_D_2} = 2\alpha C_u (d_{mg} + \beta) \frac{N-k}{(2N-k-1)T_d} . \quad (13)$$

$$U_{IP_D_2} = 2\alpha C_u (d_{mg} + \beta) \frac{1}{[1 + (N-1) \sum_{i=1}^k \frac{1}{N-i}] T_d} . \quad (14)$$

Let S_{SIP_D} and S_{DIP_D} S_{SAP_D} be the total signaling cost functions of MIP with Domain paging protocol and SIP, DIP, and SAP, respectively. They can be computed as the summation of three cost elements above as follows:

$$S_{SAP_D}(k, \lambda, T_d) = P + U_{D_{-1}} + U_{SAP_D_2} . \quad (15)$$

$$S_{SIP_D}(k, \lambda, T_d) = S_{DIP_D}(k, \lambda, T_d) = P + U_{D_{-1}} + U_{IP_D_2} . \quad (16)$$

4 Numerical Results

We now investigate and compare the performance of the paging protocols and schemes under the impact of varying parameters by evaluating the total signaling cost functions. Our performance analysis uses the parameters with numerical values listed in Table 2. This set of values represents a typical situation in a system running MIP with paging support: an MH is visiting a foreign network far away from its home network; the values of α and β are selected to capture the fact that cost weight of wireless link is higher than that of wired link.

For comparison purpose, we should consider all the four combinations paging protocols (two protocols) and paging schemes (two schemes). Fortunately, because paging protocols and paging schemes in this paper are orthogonal, we will actually investigate two combinations. More specifically, first we compare SAP scheme with Individual paging schemes (in FA paging protocol), then we investigate Domain paging protocol and FA paging protocol (both with Individual paging scheme). It

should be noted that the pure MIP is considered to be a special case of MIP with paging when paging area size $k = 1$.

In SAP, the total signaling cost function with respect to PA size k is a convex function, therefore, any PA size might be far from being optimal. In Individual Paging schemes, however, the total signaling cost functions are concave, and we can obtain the optimal PA size using the iterative algorithm in [5].

Table 2. Numerical values of parameters

N	d_{mh}	d_{mg}	d_{fp}	C_p	C_u	α	β
30	30	10	2	3.0	1.0	1.0	2.0

4.1 SAP Scheme vs. Individual Paging Scheme

4.1.1 Impact of Varying Dwell Time

Figure 5 shows the total signaling cost of Individual paging vs. that of SAP (both with FA paging) with respect to PA size k . The incoming call rate is fixed $\lambda = 0.5$ while the dwell time varies: $T_d = 0.2; 0.5; 20.0$. It can be observed that the total signaling cost of Individual paging can be much lower than that of SAP. Particularly, when the mobility of the MH is high, i.e., T_d is small, the benefit of Individual paging over SAP is significant. Up to 58% of the signaling cost can be saved by using Individual paging instead of SAP. Both paging schemes can have much better performance than that of MIP when T_d is relatively small, i.e., high mobility. But when T_d is very high such as $T_d = 20$, i.e., when the MH is almost not mobile, Mobile IP is better. This is because MIP with paging no longer enjoys its advantage of reducing location update cost due to the movement of the MH (since the MH is almost not moving) whereas the paging cost to locate it is still the same if the PA size stays unchanged. In this case, SAP scheme and Individual paging scheme are almost the same in terms of signaling cost.

In Fig. 6, the dwelling time is fixed $T_d = 0.2$ while the incoming call rate varies $\lambda = 0.4; 3.0$. It can be seen from the graphs that Individual paging schemes induce lower signaling cost than that of SAP scheme. Actually, the different element in the total signaling cost functions of them is the cost of location update due to the movement of the MH (U2) while the paging cost P and the cost of location update following a paging U1 are the same. Hence, the difference in the signaling costs of Individual paging and SAP depends on the mobility of the user, which is expressed by T_d , but not on the incoming call rate λ .

Nevertheless, when λ is relatively high, e.g. $\lambda = 10.0$, the paging cost P and the location update cost U1 become the dominating elements in the total signaling cost function, and thus, the difference between Individual paging and SAP in terms of signaling cost gets to be insignificant.

These results are well expected since each MH has its own mobility and call arrival patterns and thus has its own optimal network configuration as we argued before.

However, the benefit of individual paging comes with the price of high calculation load on the MH itself. In Static aggregate paging, the computation is usually performed by the system and based on the average values of parameters of all mobile users and the result is for all these users. In contrast, the calculation is a burden for MHs in Individual paging and it may reduce the advantage of power saving brought about by paging. The analysis results also show that the total cost function is very sensitive to PA size. Therefore, determining the optimal PA size is crucial in saving total cost of MIP with paging.

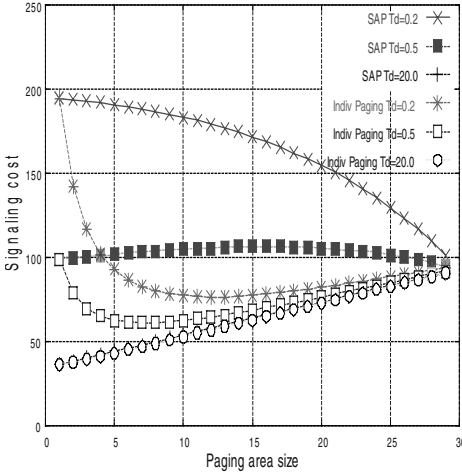


Fig. 5. Total signaling cost vs. PA size for SAP and Individual paging with varying T_d

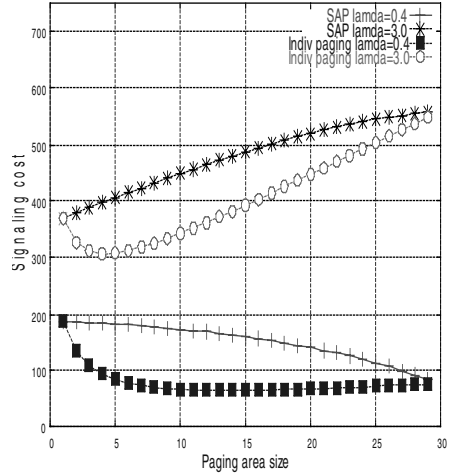


Fig. 6. Total signaling cost vs. PA size for SAP and Individual paging with varying λ

4.2 Domain Paging Protocol vs. FA Paging Protocol

4.2.1 Impact of Varying Dwell Time

In Figure 7, the incoming call rate is fixed $\lambda = 0.5$ while user dwell time varies $T_d = 0.2; 0.5$. Domain paging is shown to offer considerably lower signaling cost than that of FA paging, and the difference is even bigger when user mobility is high, i.e. low dwell time.

4.2.2 Impact of Varying Incoming Call Rate

The signaling cost of MIP with Domain paging protocol and that of MIP with FA paging protocol (both with Individual paging scheme) are shown against paging area size k in Fig. 8. T_d is fixed and equal to 0.2 while λ varies: $\lambda = 0.8; 3.0$. From the figure, the total signaling load of MIP with Domain paging is significantly lower than that of MIP with FA paging, especially when the incoming call rate is high.

The advantage actually comes from the benefit of domain-based localization of location registration of mobile users in MIP with Domain paging. Our results demonstrate that Domain paging can offer up to more than 50% cost saving compared with FA paging. The gain of Domain paging over FA paging depends on both user dwell time T_d and incoming call rate λ . This result is explicable from the formulas of their total signaling cost functions. The different elements of the two protocols are U1 and U2, which are functions of λ and T_d , respectively.

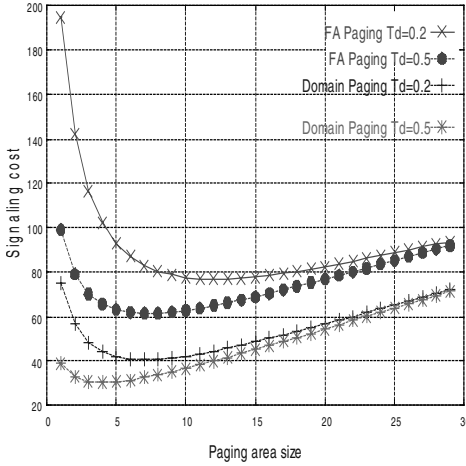


Fig. 7. Total Signaling Cost vs. PA size for Domain paging and FA paging with varying T_d ($\lambda = 0.5$; $T_d = 0.2; 0.5$)

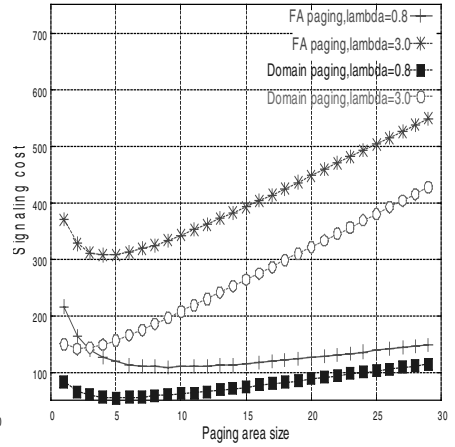


Fig. 8. Total Signaling Cost vs. PA size for Domain paging and FA paging with varying λ ($T_d = 0.2$; $\lambda = 0.8; 3.0$)

5 Conclusions

In this paper, we have investigated the performance of different paging protocols and schemes with MIP, based on their total signaling cost functions. We have also shown how to construct the optimal PA and how important it is in saving the signaling cost of MIP with paging. Our analysis results show that paging can be an effective approach to improve MIP. Up to 60% cost saving can be acquired. However, paging service is not a good choice for very low mobility users. Individual paging can significantly outperform SAP and the cost saving may be as high as 58%. The results also demonstrate that Domain paging can enjoy much lower signaling cost than that of FA paging (up to 50% cost saving). The impact of various user parameters on the total signaling cost is also investigated thoroughly in our analysis.

In our future works, we will consider other mobility models and assumptions. Specifically, we will investigate more general patterns of call arrival and call duration such as Poisson distribution of incoming calls. This also includes the case of over-

lapped incoming calls. Other mobility models such as Random Walk and Markov models will be taken into account.

References

1. R. Ramjee, L. Li, T. L. Porta and S. Kasera, "IP Paging Services for Mobile Hosts," *Wireless Network* 8, 427–441, 2002.
2. C. Castellucia, "Extending Mobile IP with Adaptive Individual Paging: A Performance Analysis," *MobilCom*, Volume 1, Number 2.
3. X. Zhang, J. Castellanos, and A. Campbell, "Design and Performance of Mobile IP Paging," *ACM Mobile Networks and Applications (MONET)*, Special issue on Modeling Analysis and Simulation of Wireless and Mobile Systems, Vol. 7, No. 2, March 2002.
4. J. Xie, and Ian F. Akyildiz, "A Novel Distributed Dynamic Location Management Scheme for Minimizing Signaling Costs in Mobile IP," *IEEE Trans. on MobilCom*, Vol. 1, No. 3, Jul-Sept 2002.
5. H.Xie, S. Tabbane, and D. J. Goodman, "Dynamic Location Area Management and Performance Analysis," *Proc. 43rd IEEE VTC*, pp. 536–539, 1993.
6. J. Kempf, and P. Muta, "IP Paging Considered Unnecessary: Mobile IPv6 and IP Paging for Dormant Mode Location Update in Macrocellular and Hotspot Networks," *IEEE Wireless Communications and Networking Conference*, New Orleans, Louisiana, USA, March 2003.
7. C. Perkins and D. Johnson, "Route Optimization in Mobile IP," Internet Draft, Internet Eng. Task Force (IETF), Nov. 2000.
8. Young J. Lee and Ian F. Akyildiz, "A New Scheme for Reducing Link and Signaling Costs in Mobile IP," *IEEE Transactions on Computer*, Vol. 52, No. 6, June 2003.

Preventive and Corrective Protection for Mobile Ad Hoc Network Routing Protocols

Ricardo Staciarini Puttini¹, Ludovic Mé², and Rafael Timóteo de Sousa Jr.¹

¹ University of Brasilia, Department of Electrical Engineering, CP 4386,
70919-970 Brasília, Brazil
{puttini, desousa}@unb.br

² École Supérieure d'Électricité – Supélec, BP 81127
35511 Cesson-Sévigné Cedex, France
{ludovic.me}@supelec.fr

Abstract. In this paper we describe vulnerabilities and possible protections for mobile ad hoc networks (MANET) routing protocols. Vulnerability and adversary models are built to describe impersonation, fabrication and modification attacks. A security model is proposed, considering both preventive and corrective protection. The basic preventive protection consists of a certificate-based authentication mechanism, which is designed as a MANET authentication extension (MAE) that provides authentication for all routing protocol messages. Corrective protection consists of an intrusion detection and response service (IDS). Certification service and IDS are both provided in a distributed and self-organized manner. Intrusion response is mainly defined in terms of interaction between certification service and IDS. The proposed vulnerability analysis and security design are detailed and validated using the Optimized Link State Routing (OLSR) Protocol.

1 Introduction

In this paper we propose a new security model for protection of MANET routing protocol. The salient features in our design are:

- Combination of both preventive and corrective protection;
- Self-organized conception of security services, in the sense that security services are provided collaboratively, without assumption on any centralized entity;
- Fully localized solutions, restricting communication overheads within nearby nodes; and
- Robustness in the presence of node compromising, combining both preventive and corrective security services.

As a basic preventive solution, a digital certificate based authentication service is proposed for the routing protocol messages. The companion certificate services are also proposed, as an extension to [1], which is designed to be self-organized and fully localized. An intrusion detection and response system (IDS) provides the corrective solution feeding the certification service with information about misbehaving nodes, which are eliminated from the network by certification revocation. The proposed

model is completely developed for protection of the Optimized Link State Routing (OLSR) Protocol¹. Validation of the proposed model is obtained from actual implementation of security services for the OLSR.

The rest of this paper is organized as follows: Section 2 discusses related work. In section 3, we discuss the vulnerability and adversary models defined in our proposal. Section 4 is devoted to description of the security model. Section 5 details the development of the proposed solution for protection of the OLSR. Section 6 describes the implementation and results obtained from experimentation. Finally, section 7 concludes the paper with our final remarks.

2 Related Work

Most of the current research in MANET security is devoted to provision of preventive protection for the routing protocol, usually by means of an authentication service similar to ours [2,3,4]. Alternative approaches are based in the establishment of security associations between nodes, allowing the use of symmetrical cryptography instead of public key cryptography. These associations may be derived from node synchronization such as in [5] or directly from mobility, allowing local security associations only [6]. As a general rule, these solutions are not tolerant to the presence of malicious or compromised nodes in the MANET.

On the other hand, research results on intrusion detection in MANET have only started to appear. Also, published intrusion detection approaches do not address intrusion response yet. This is the case for [7,8], where basic MANET IDS architectures have been proposed and preliminary results were presented. An intrusion detection strategy to deal with non-cooperative nodes in ad hoc networks is presented in [9]. However, there isn't any notion of collaborative security services in this approach.

The work in [10] proposes an intrusion-tolerant security solution for the AODV protocol. However, the designed solution doesn't incorporate any preventive (authentication) protection. Instead, only a simple neighbor verification mechanism is used. Unfortunately, this mechanism is based in an erroneous assumption that MAC address cannot be spoofed. Moreover, the intrusion detection mechanism limited only to RREP message flooding, which do not generalize to accomplish all the attacks described in terms of fabrication, modification and impersonation of other routing protocol messages.

3 Vulnerability and Adversary Models

3.1 Vulnerability Model

Attacks against routing protocols are usually related to the insertion of erroneous routing information, attempting to disturb the routing algorithm. This is the case for

¹ OSPF, AODV, TBRPF and DSR routing protocols are specified in experimental RFCs, which are available from IETF at <http://www.ietf.org/html.charters/manet-charter.html>.

modification (malicious modification of routing protocol messages), impersonation (masquerading as another node) or fabrication (generation of false routing messages) attacks. Combinations of these basic operations are also possible and provide a broader range of attacks. There are also some cases where passive eavesdropping vulnerabilities may also be considered (e.g. in military application, where the routing protocol messages can reveal information about geographical positioning of the nodes). Additionally, trivial attacks based in resource consumption and non-cooperation are possible in all ad hoc routing protocols. In this paper, we focus on vulnerabilities related to impersonation, modification and fabrication of routing protocol messages.

Each node in MANET keeps local routing information in order to provide the routing service. Nodes use routing protocol messages to share such local routing information. We define an “adversary” as any node announcing erroneous routing information in fabricated, modified and/or impersonated routing protocol messages. Also, a “target” is any node accepting and using this erroneous information.

We admit that modified/fabricated messages have valid syntax. Adversaries may exploit any message defined as mandatory for the routing protocol. If a message is fabricated, the adversary should either masquerade as some node that is already present in the network or use any unallocated network address.

3.2 Adversary Model

Although it might seem that the MANET routing protocol vulnerabilities considered here are quite similar to those from classical routing protocols [11], exploitation of such vulnerabilities are quite different in the MANET, given the particular features of these networks [12]:

- promiscuous nature of the wireless link (adversary may promiscuously listen to wireless transmissions);
- non-centralized, peer-to-peer communication model/lack of infrastructure (adversary may communicate directly with any node within the transmission range of its wireless interface); and
- mobility and dynamic network topology (adversary may move with limited speed to gather information about other nodes or to escape from intrusion detection).

Moreover, unlike classical routers, which provide only limited service with careful protection, MANET nodes have a non-negligible probability of compromise due to vulnerabilities related to OS, software bugs, backdoors, viruses, etc. Also, a mobile node without adequate physical protection is also prone to being captured. Although we do not elaborate on such vulnerabilities, we admit that an adversary may be able to compromise or capture a mobile node. We do not restrict the consequences of a node break-in. Thus, during break-in, any secret information (including private or shared keys) stored locally may be exposed to the intruder. Any broken node may be either used to launch routing protocol attacks or may be impersonated. As there is no way to distinguish between these situations, we do not differentiate compromised nodes from adversaries, from the security point of view. Neither do we differentiate insider from outsider adversaries.

4 Security (Protection) Model

MANET context imposes strong requirements in the protection model. The MANET requirements considered in our security model are:

- **Mobility:** nodes in a MANET may, at any time, disappear from, appear into or move within the network. Therefore, availability of an individual node cannot be assured security services cannot rely on a central entity.
- **Locality:** the error prone nature of the wireless links and the limited bandwidth requires that security services must be provided collaboratively by nearby nodes, most often by 1-hop neighbor nodes.
- **Intrusion Tolerance:** security solution should be robust in the existence of compromised nodes in the network, given the non-negligible probability for node break-ins.

To cope with mobility of the MANET nodes, we do not assume in our design the existence of any centralized entity in the network. Instead, we take the self-organized approach by adopting fully localized mechanisms and relying on the collaboration for the provision of the security services. An autonomous instance of each security service must be active in each MANET node. These instances are generally called Local Service (L-Service). A L-Service collaborates with L-Services from nearby nodes (usually in the neighborhood), by means of some collaboration protocol. This sense of self-organization is exactly the same used in the very conception of the MANET routing service, the L-Service being represented by the MANET routing protocol daemon, which is autonomously executed in each MANET node, the collaboration protocol being represented by the MANET routing protocol.

In our design, protection of the routing protocol includes both preventive and corrective security services. A certificate-based authentication service for the routing protocol messages is considered as a basic preventive solution. The authentication service aims to avoid an attack to be generated from a non-authenticated node. However, according to the presumed adversary model (section 2.2), attacks are still possible in two situations: (1) an authenticated node (e.g. certificate holder) starts to behave maliciously; or (2) a MANET node has been compromised and the authentication secret (e.g. private key) from that node has been exposed. The corrective security service is provided in terms of an intrusion detection and response system (IDS). Intrusion response consists mainly in the isolation of compromised nodes, excluding them from the routing service. This is accomplished by means of certificate revocation.

Certification services and intrusion detection and response should be provided in a self-organized and distributed manner by a Local Certification Service (L-Cert) and a Local IDS (LIDS) instances [8]. Fig. 1 illustrates the proposed protection model. Basically, routing protocol, certification service and IDS (alert) message exchanges must be authenticated with a MANET authentication extension (MAE), which is appended to each message and provides the authentication information. Authentication is based on the certification service and uses asymmetric cryptography primitives. Each node in the MANET must hold a valid certificate, binding the node's identity to its public key.

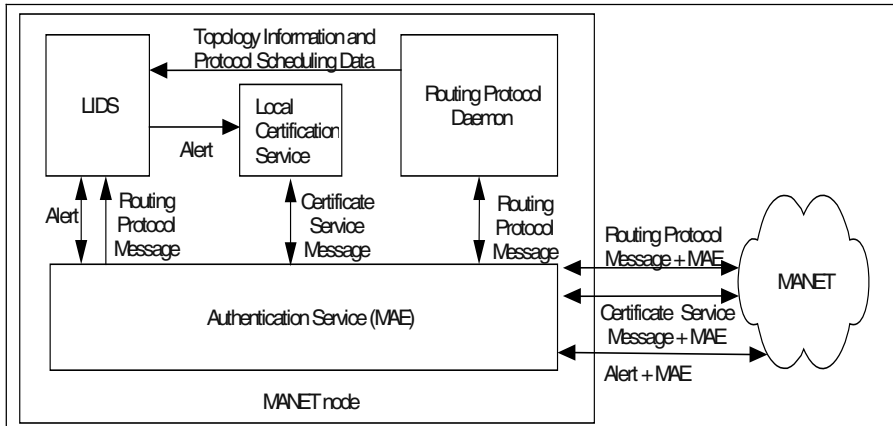


Fig. 1. Routing Protocol Protection Model

Whenever a node is broken, an adversary becomes able to impersonate the compromised node and may fabricate/modify not only routing protocol messages but, also, certificate service messages and alerts (IDS message). In order to maintaining the robustness of the security solution in the presence of compromised nodes, security services in our system are designed to have k -by- n security, in the sense that any certification service or intrusion response must be collaboratively provided by, at least, k nodes, where n is the total (non-fixed) number of nodes in the network. Thus, for compromising a security service, an adversary must break into k different nodes. Correct nodes running the LIDS must detect the attacks against the routing protocol and isolate the compromised node (by revoking its certificate) before that an adversary can compromise k nodes, breaking the collaborative security system.

Given that collaboration is done by means of authenticated messages (certificate services and IDS messages are also authenticated with MAE), isolating a node is equivalent to revoking the node's certificate. An indirect revocation mechanism is related to certification expiration. Thus, security can be improved if we require that certificates must be renewed from time to time. Certificates are issued with constrained certificate expiration time. Each node having a valid certificate must request for a new certificate, before the current certificate has expired. Nodes that are not well behaving should not have their certificates renewed.

Finally, locality requirement states that collaboration should be designed to restrict communications among L-Services (e.g. local certificate services and LIDSes) around nearby nodes, usually in the local neighborhood. This is an important requirement as it relates to the scalability of the overall solution. Considering the locality requirement, k becomes an important parameter and should be related to the average size of neighborhoods in the network. If a node has k or more neighbors, IDS and certification services can be fully provided in the local neighborhood. Thus, the security solution is scalable, in the sense that security services are run locally, provided a convenient choice for the parameter k .

4.1 MANET Certification Service

The design of self-organized certification services in MANET has been discussed in a few recent papers [1,13], which are based on a distributed certification authority (DCA) trust model. The CA secret key (K_{CA}) is used to sign certificates for all nodes in the MANET. A certificate signed with K_{CA} can be verified with the well-known system public key. The distribution of the CA capabilities is achieved by sharing the secret key among network nodes by means of threshold cryptography techniques [1]. Each MANET node holds a private-key-share (SK_{CA}) and any k (a system wide constant, usually related to the average number of neighbors) of such private-key-share holders can collective function as a CA. The K_{CA} , however, is not recoverable by any node. Counter-certificate issuing does certificate revocation, which must also be signed with K_{CA} . Our proposal is based on [1], with improvements in certification policy specification, local certificate management and multiple DCA support[14].

4.2 Authentication in MANET Routing Protocols

The authentication service considered in our model is provided by a MANET authentication extension (MAE), which is appended to each routing protocol message or packet. This MAE contains all the authentication information required to correctly assure authenticity and integrity to the message or packet being protected. Our objective is to design such extension in a flexible and adaptable way, so that it could be used to secure different MANET routing protocols. The idea is to preserve the routing protocol message syntax unchanged, differently from previous work [2-4].

Authentication Objects: MAE is composed by authentication objects. At least one (mandatory) authentication object should be present in the MAE and alternatively contains a message authentication code (MAC) object, which is computed as a hash-function applied to the data being authenticated keyed with a private-key-shared team key, or a digital signature (DS) object. MAC/DS authenticates all the non-mutable fields of a routing protocol packet/message. Additional authentication objects are used to provide optional services. Currently defined options are signer certificate (using to carry the certificate of the MAE signer within the message), hash chains information (keeping additional authentication data related to protection of mutable fields in the packet/message of AODV and DSR) and sequence number (for reply protection) [14].

Mutable Fields: In DSR and AODV, MANET routing protocols there are messages that progressively change while they are forwarded by intermediate nodes in the path between message source and destination. While it is desirable that this mutable information should also be protected, such protection usually implies in increasing the authentication information size each time the message is forwarded. This is not surprising as the information contained in the message is due to all nodes that have previously forwarded it and, each of them should be authenticated in general. Some methods to protect typical mutable information (e.g. hop count, IP address based routing trace, etc.) have been proposed [15] and may be used in our design.

4.3 MAE for DSR, AODV, OLSR, and TBRPF

OLSR and TBRPF are proactive link-state routing protocols whose message don't have mutable fields in the routing messages that are actually used by the respective routing protocol algorithm. MAE for securing OLSR and TBRPF is simply built with a single authentication object containing MAC or digital signature (DS).

AODV have mutable fields in route request (RREQ) and route reply (RREP). These fields are hop count metrics that are changed every time the packet is processed and forwarded by nodes between the message source and destination. A hash chain object (HC) [4] is included and updated each time these fields change (e.g. each time the message is processed and forward). Such protection avoids that an attacker could decrement the hop count. Route error (RERR) messages are signed only by the node forwarding them. Route reply acknowledgments (RREP-ACK) have no mutable fields and are only authenticated by the message originator.

Securing DSR is quite more complex, although limited security can be achieved by combining the mandatory authentication object with a hash chain object implementing a per-hop hashing schema in RREQ messages. This will avoid an attacker from faking of the initiator node and from removing correct IP address in the route list [5]. RREP messages could be simply signed by the target of the route discovery (e.g. the node originating the RREP message).

Table 1 illustrates the main features of each MANET routing protocol and MAE requirements for each of them.

Table 1. MAE for MANET routing protocols

Routing Protocol	Routing Discovery	Routing Algorithm	Relevant Messages	Authentication Objects
DSR	on-demand	source-routing	RREQ	DS+HC
			RREP	DS
AODV	on-demand	distance-vector	RREQ	DS+HC
			RREP	DS+HC
			RERR	DS
			RREP-ACK	DS
OLSR	proactive	link-state	Hello, Topology Control	DS
TBRPF	proactive	link-state	Hello, Topology Update	DS

4.4 Collaborative Intrusion Detection and Response

Present intrusion detection concerns are usually divided in three main processes: data collection, detection algorithm design and alert management. A simple IDS model consists of three modules: Sensor, Analyzer and Manager, each of them being related with one of the intrusion detection processes. More precisely, a Sensor collects data from a data source, an Analyzer processes the collected data for detecting signs of events that might have security concerns and the Manager stands for the management interface of whole process, besides of doing alert correlation and response initiation.

Given the lack of centralization, the mobility of the nodes and the wireless nature of link connections in the MANET environment, some (if not all) of the tasks required for the intrusion detection process described above should be executed in a distributed and cooperative manner [7,8]. To active these objectives, the MANET-adapted IDS is designed with the following features: (1) each MANET node runs an autonomous instance of a local IDS (LIDS); (2) each LIDS is functionally complete, in the sense that it may execute the whole detection process (e.g. data collection, detection algorithm execution and alert management); (3) LIDS collaborate with each using a mechanism that takes into account the restrictions resulting from the MANET context; e.g. limited bandwidth or poor connectivity.

Fig. 2 shows the proposed architecture for the LIDS. Besides of the basic IDS functional modules (e.g. Sensor, Analyser and Manager), Distribution Manger and LIDS Cooperation Protocol are also included in the architecture, in order to cope with the distribution and cooperation requirements.

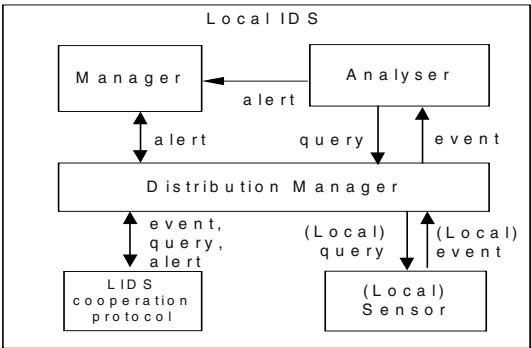


Fig. 2. LIDS Architecture

4.5.1 Sensor: Data Sources

In our process, the data collected for intrusion detection consists of all routing protocol messages, which are obtained from the authentication service. The sensor also maintains information about the neighborhood topology and the protocol message scheduling, which are used to extract information from a new received message that could be relevant to the detection process.

4.5.2 Analyzer: Intrusion Detection Algorithm

The Analyzer processes the events according to some defined detection strategy. At least two detection methodologies are currently in discussion: misuse and anomaly detection. Misuse detection relates to the identification of patterns (e.g. event sequences) that characterizes a known attack type, which are called attack signatures. Alternatively, anomaly detection consists in characterizing the normal system behavior and detecting deviations from this normal pattern. In our model, we use the misuse intrusion detection strategy. The principal advantage of the misuse approach relates to the possibility of identification of the attack type being detected and even, in some cases, the identification of the attack source. This last feature is required in our de-

sign, as intrusion detection is used to identify misbehaving nodes that must be isolated. In misuse IDS, attack signature should be supplied for each attack (or class of attacks) that must be detected. Attack signatures can be generally described by patterns that become observable when the attack is launched. In the case of modification, fabrication and impersonation attacks against the routing protocol, these patterns correspond to anomalies in the scheduling of the routing protocol or inconsistencies in the routing information advertised simultaneously by different nodes.

4.5.3 Collaboration in the Intrusion Detection

The Distribution Manager module receives all IDS messages (e.g. event, query and alert), either if the message was locally generated or received from remote nodes, and decides if the message should be consumed locally or if it should be dispatched to a remote node. The IDS Cooperation protocol module implements the communication aspects of the cooperation.

Data collection is always local but relevant events may be communicated to other remote nodes in order to help them in the intrusion detection processing. Also, if a LIDS needs to know about an event that may be occurring in a remote node, it can query the remote node by sending a query message.

4.5.4 Collaboration in the Intrusion Response

As a general rule, each node must monitor the behavior of neighbor nodes. LIDS executes this monitoring. If an adversary launches an attack against the routing protocol, correct neighbors receiving the faked routing protocol message may possibly detect the attack.

The nodes detecting the attack collaborate with its neighbors to provide intrusion response by signing an accusation (alert) against the detected adversary. This alert is also sent to the local certification service, which signs a partial counter-certificate for the adversary. Partial counter-certificates are flooded in the network.

Correct nodes may collect alerts from different nodes detecting and attack. A node collecting, at least, k accusations against the same adversary will also sign a partial counter-certificate for it, even if the node haven't detect any attack coming from that adversary by itself.

Redundancies in the MANET should compensate for the nodes that are not cooperating in the detection and response processes. Indeed, it will be shown that it is possible for more than one single node to track and detect the same attack. If any combination of k nodes in the network detects an attack coming from the same adversary, the adversary's certificate will be revoked.

5 OLSR Vulnerability and Protection Analysis

5.1 OLSR Background

OLSR operates as a table driven proactive routing protocol, which means that it is based on the regular exchange of network topology information between nodes. The topological information is used for updating the routing table of participating nodes

by means of a link-state routing algorithm. The routing metric is always hop-distance. Thus, the protocol gives minimum hop distance routing when the network is in a stable state. Optimization over a pure link state algorithm is obtained by reducing the size of control messages and minimizing flooding of control traffic, which is executed only by some selected nodes called MPR (Multi Point Relays). OLSR communicates using a unified packet format for all data related to the protocol. Each packet is carried in UDP and contains one or more OLSR messages.

The nodes use HELLO messages to detect and update their neighbor set. Each node periodically broadcasts HELLO messages, containing information about heard neighbor interfaces and their link status. The link status may either be “symmetric” (link has been verified to be symmetrical), “heard” (link is asymmetrical), “MPR” (node is selected as MPR, link must also be symmetric) or “lost” (neighbor have moved away). HELLO messages are periodically broadcasted from each node to all 1-hop neighbors and emitted on each MANET interface of the node. These messages are not relayed to other nodes.

Each node in the network independently selects its own MPR set among his “symmetric” neighborhood. The MPR set must be computed by a node in such a way that, through the neighbors in the MPR set, it can reach all symmetric 2-hop neighbors, which are not at the same time symmetric neighbors of the node.

For provision of routes to faraway nodes, each node maintains topological information about the network. This information is acquired by means of OLSR topology control (TC) messages and is used for routing table updates. Nodes that have been selected as MPR by other nodes periodically generate the TC messages, which contain the list of all selector nodes (MS). TC messages are flooded to the whole network by the MPR nodes. A “Message Sequence Number” field is used to avoid duplicated message processing.

5.2 OLSR Vulnerabilities

The attacks being described here rely on the fabrication of OLSR HELLO and TC messages or on modification of OLSR TC messages. All attacks basically have denial-of-service (DoS) effects. Table 2 summarizes the attack identification following the vulnerability model described in section 3.

Table 2. OLSR Attack Identification

Attack	OLSR Message	Disrupted Information	Message Originator Identification	Attack Signature
Fabrication	HELLO	Neighbor List		Inconsistency in routing information
Fabrication + Impersonation	HELLO	Link-status	IP address of target node	Anomaly in the scheduling
Fabrication	TC	MS list		Inconsistency in routing information
Modification + Impersonation	TC	Sequence Number	IP address of target node	Anomaly in the scheduling

5.3 OLSR Message Authentication

None of the OLSR messages (e.g. HELLO, TC, MID, HNA and FRR) has any mutable fields in the message data. However, each message has a message header, which contains a “hop count” and a “time to live” mutable fields. HELLO and FRR messages are broadcasted only in the originator neighborhood, while TC, MID and HNA messages are flooded in the whole network. Given that these fields are not used in the routing table calculation but only in the flooding algorithm (which is robust by itself, provided that there are redundancies in the network topology), no additional protection is required for authentication of the mutable fields. Thus, OLSR MAE consists of a single digital signature, authenticating all fields in message data and in the message header, except from the “hop count” and “time to live” fields, which must be zeroed for the digital signature computation.

5.4 OLSR Intrusion Detection

OLSR intrusion detection is accomplished by implementation of Sensor and Analyzer modules that must, respectively, collect information related to the attacks and analyze the information searching for occurrences of patterns representing signatures for each one of the attack. Whenever detecting an attack, the Analyzer generates the respective alert and pass it to both Manager and Distribution Manager modules, which will collaborate with other nodes to provide the intrusion response. The collected information (Sensor) consists of all HELLO and TC routing messages and some topological information maintained by the routing daemon.

Information analysis (Analyzer) is done whenever a new HELLO or TC message arrives and consists in the identification of the attack signature as described below:

- Attack 1: This attack can be characterized by identification of inconsistency in routing information from different HELLO messages. Nodes that hear HELLO messages from both the attacker and correct nodes announced in the fake message will detect the attack by verification of inconsistencies in these messages.
- Attack 2: This attack can be characterized by the anomaly in the scheduling of routing messages related to the reception of both correct and spoofed messages with the same originator information and advertising the link type of some neighbor as “lost” and as “symmetric” in the same HELLO_INTERVAL period.
- Attack 3: This attack can be characterized by the presence of inconsistencies in the routing information advertised simultaneously by different nodes. Fake TC message are flooded in the network and these messages will eventually arrive at the nodes being advertised as MS and at their neighbors. These nodes detects the attack, as advertised nodes don’t have the adversary in their neighbor set.
- Attack 4: This attack can be characterized by the anomaly in the scheduling of routing messages. The actual originator node and its neighbors, which receive both correct and modified TC messages, can detect the attack by verifying the occurrence of TC messages from the same originator, advertising the same MS set but with different “message sequence number”, during the same TC_INTERVAL period.

6 Implementation and Results

The MAE and the local certification service were implemented along with the available implementation of OLSR v.3. The openssl library was used for the cryptography routines. The LIDS was coded separately, and mobile agents were used for collaborative intrusion detection [8]. Attacks described above were implemented by using the tcpdump packet capture library (libpcap).

The developed platform was tested in an experimental MANET with 10 nodes running on Linux/Intel laptops with IEEE802.11b cards. Two of them are playing the role of adversary nodes. The number of nodes in service coalition was fixed to $k = 3$ in all experiments. Certification renewal was required at each 60 minutes.

6.1 Computational and Network Performance Considerations

Overhead of the proposed protocols has been preliminarily evaluated through our experiments with the OLSR implementation. Considering the network overhead, a MAE transmitted without certificates have a fixed size of 72 bytes, for an RSA key of 512 bits. Average size of OLSR messages depends on the network size and density. For example, in a 100 nodes MANET, which are uniformly distributed over a 1000m x 1000m area and having a transmission range of 200 m, the average size of a HELLO message is 64.26 bytes (each node having an average neighborhood of 12.56 nodes). The high overhead represented by the MAE is due to the use of asymmetric cryptography. In our experiments the message size observed were comparatively smaller, because our real MANET had only 10 nodes. In any case, an OLSR packet containing a HELLO or TC message and a certificate loaded MAE do not oversize the 512-byte packet limit of the OLSR implementation.

LIDS network overhead were limited to alert propagation during detection of any attack in the neighborhood of the node detecting the attack.

Computational overhead of the authentication service was analyzed indirectly by evaluation of RSA signature generation and verification. In the MAE verification process, two signatures may be verified, if the MAE signer certificate is not cached and must be validated. Time for executing a RSA signature generation and verification (512-bits key) were averaged in a Pentium III (900MHz, 128M RAM, running Red Hat Linux with kernel 2.4.7) to 9ms and 2.6ms, respectively. The normal OLSR packet processing (packet reception) was estimated in 2.5ms. Storage requirements of our proposal are mainly related to certificate cache storage (as CRL can not oversize k , a small constant). If all certificates in the 100 nodes MANET being simulated were locally cached, a 26kbyte cache is due, which is perfectly reasonable.

6.2 Security Evaluation

Attacks have been successful in corrupting routing when authentication was disabled for the routing protocol. All four attacks were played by two adversary nodes. Attack effects were analyzed for this topology in three different scenarios: (1) with no pro-

tection at all; (2) with only preventive protection (authentication); and (3) with both preventive and corrective (IDS) protections.

In the first scenario, routing disruption was readily obtained and persisted while the adversaries continued to send the fake messages. In the second scenario, the adversaries needed to have a valid certificate to authenticate messages, in order to successfully realize the attacks. This is equivalent of the compromising of some MANET node. If the attacks were played with valid authentication information, the same results that have been observed in scenario 1 for the routing disruption were observed. Finally, in the third scenario, the attack effects on routing disruption were completely mitigated, all the attacks being detected, with no false negatives, by at least 3 nodes (neighbors from the adversaries nodes) that had collaborated to isolate both the adversaries.

Another important issue concerns the choice for the k parameter. Clearly, there is a tradeoff between security and performance/availability in this choice. If k is chosen to be lesser than the neighborhood size, all services are locally provided. However, if there are compromised nodes it is possible that there isn't enough correct nodes in the neighborhood for local intrusion detection. In our experiments, we have chosen $k = 3$ because our neighborhood size is 5, so even in the presence of 2 compromised nodes (the maximum number of compromised nodes allowed in this security solution), we shall have at least 3 correct nodes (the minimum number of nodes required to detect the attack).

The initial certificate distribution in our experiments was done out-of-band but certification renewal automatically executed at each 60 minutes. As long as the correct nodes detect both misbehaving nodes, these cannot renew their certificates.

7 Extension to Other MANET Routing Protocols

OLSR and TBRPF messages do not have any mutable fields that are directly used by the routing algorithm, and so, the authentication data in the MAE for these protocols is a single digital signature. MAE for AODV and DSR must provide additional data to authenticate the mutable fields of these protocol messages, such as additional digital signature (signed by nodes modifying and forwarding the original message [2]) or hash chains [3,4].

LIDS design must be carried out for considering the particular features and vulnerabilities of each MANET routing protocol. More specifically, attack signature should be identified for each routing protocol vulnerability. Nevertheless, the IDS architecture should be effective in any case.

8 Conclusions

We have presented in this paper a novel security model for MANET networks that incorporates both preventive and corrective protections. The security services designed in our proposal are self-organized and have shown to restrict communication and processing overhead among sets of few nearby nodes.

In our approach, vulnerability analysis considers the intrusion detection by defining attack signatures due to anomalies in topology and routing protocol scheduling. The security solution uses both preventive and corrective protections and security services are designed to be self-organized.

Finally, the usage of the same authentication service (MAE) for both routing protocol and security service messages was successful, providing some insights for future research on the preventive protection of the security service messages.

References

1. J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, "Providing robust and ubiquitous security support for MANET," IEEE ICNP 2001, 2001.
2. B. Dahill, K. Sanzgiri, B. N. Levine, C. Shields and E. Royer, "A secure routing protocol for ad hoc networks". In the Proceedings of the 2002 IEEE International Conference on Network Protocols (INCP 2002), Nov. 2002.
3. P. Papadimitratos and Z. J. Haas. Secure routing for mobile ad hoc networks. SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNSD 2002), Jan 2002.
4. M. Guerrero and N. Asokan, "Securing Ad Hoc Routing Protocols", in the Proceedings of 2002 ACM Workshop on Wireless Security (WiSe'2002), in conjunction with the ACM MOBICOM2002, September, 2002.
5. Y. C. Hu, A. Perrig, and D. Johnson, "Ariadne: A secure On-demand routing protocol for ad hoc networks", in the Proceedings of ACM MobiCom 2002, Sep. 2002.
6. S. Capkun, J.P. Hubaux, L. Buttyán, "Mobility helps security in ad hoc networks", Proceedings of the fourth ACM international symposium on Mobile ad hoc networking & computing (MobiHoc 2003), pp. 46–56, 2003.
7. Y. Zhang and W. Lee – Intrusion detection in wireless ad hoc networks. Proc. of 6th Annual Int. Conf. on Mobile Computing and Networking, pp. 275–283, 2000.
8. Puttini, R; Percher, JM; Me, L, Camp, O; de Sousa, R. "A Modular Architecture for a Distributed IDS for Mobile Ad Hoc Networks". Lecture Notes on Computer Science vol. 2669, Springer-Verlag, pp. 91–113, 2003.
9. K. Bradley, S. Cheung, N. Puketza, B. Mukherjee and R. Olsson - Detecting disruptive routers: a distributed network monitoring approach, Proceedings of the IEEE Symposium on Security and Privacy, pp. 115–124, 1998.
10. H. Yang, X. Meng and S. Lu, "Self-Organized Network Layer Security in Mobile Ad Hoc Networks", in the Proc. of ACM Workshop on Wireless Security (WiSe 2002), 2002.
11. F. Wang, F. Wu – On the vulnerabilities and Protection of OSPF Protocol. Proceedings of 1998 International Conference on Computer Communications and Networks, 1998.
12. S. Corson and J. Marker – Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation consideration. RFC 2501 (informational), IETF, 1999.
13. L. Zhou and Z. J. Haas. Securing ad hoc networks. IEEE Network Magazine, 13(6):24–30, November/December 1999.
14. R. Puttini, L. Me, R. de Sousa, "MAE – MANET Authentication Extension for Securing Routing Protocols", in Proc. of the 5th IFIP Int. Conf. on Mobile and Wireless Communications Networks (2003).
15. Y.C. Hu, A. Perrig, D. Johnson, "Efficient Security Mechanisms for Routing Protocols", Proceedings of the 2003 IETF Network and Distributed System Security Symposium (NDSS 2003), 2003.

Dynamic Service Adaptation for Runtime System Extensions

Robert Hirschfeld, Katsuya Kawamura, and Hendrik Berndt

DoCoMo Communications Laboratories, Future Networking Lab,
Landsberger Strasse 308-312, 80687 Munich, Germany
{hirschfeld, kawamura, berndt}@docomolab-euro.com

Abstract. Most of all software systems have to be changed after their initial deployment. This is not only because of changing knowledge and expectations about our domains and systems, but also because of the continuous change of the environment itself. While changes in the environment happen implicitly, we need to explicitly keep our technology in sync with the changing world around it. This is especially true for next generation mobile communication systems which we expect to be open to third-party service providers, allowing them to offer services on a variety of service platforms. Not all of these services to be offered will match with all of the platforms. Adjustments and extensions need to be made to offer a pleasant service experience. Research on dynamic service adaptation provides concepts and technologies needed to perform such changes late in a system's lifecycle, possibly on demand, at runtime, without disruption of service.

1 Introduction

Our research at DoCoMo Euro-Labs is directed towards mobile communications technologies beyond the third generation (B3G) that can respond to the requirements of a highly developed multimedia age. B3G systems are expected to not only integrate several access technologies, but also to promote a significant wealth of services offered by a multitude of service providers. In addition to seamless and secure access to heterogeneous networks, B3G systems are considered to encompass high service availability and best service quality to the end user. System requirements are highly demanding. Some of the key requirements essential to B3G communication platforms are to shorten development and provisioning cycles, to minimize system downtimes, to support runtime updates and upgrades, to allow for third-party service integration, and to assist in service personalization.

The unanticipated nature and complexity of forthcoming services and applications makes the support of dynamic service adaptation (DSA) and unanticipated software evolution (USE) inevitable. We regard DSA to be part of the foundation to address phenomena of USE. DSA is motivated by our continuously changing environment, a heterogeneous service landscape, as well as an open system infrastructure. Major goals of DSA are to enable service and platform evolution, to support the advancement of individual parts at a different pace, and to facilitate personalization, context-awareness, and ubiquitous computing. Mobile communication systems that

can be described as long-lived, continuously running, highly-available, embedded, or large-scale widely distributed are most suitable candidates to benefit from DSA.

Most of the adaptation mechanisms deployed today concentrate on content, a few on communication, but almost none on service logic or behavior itself. Thus, content as well as communication adaptation is understood much better than that of service logic or service behavior. In this paper the terms service adaptation, service logic adaptation, or service behavior adaptation are used interchangeably. In contrast to more traditional approaches, we combine aspect-oriented programming with computational reflection and late binding to adapt services and service platforms when changes actually require doing so, as late as possible, if possible without disruption of service.

In this paper we give an overview on our research on software engineering principles and mechanisms for DSA allowing us to evolve, adapt, and extend services dynamically to better support seamless service provisioning and application integration for the next generation mobile communication systems. Our work is aligned with active research in the field of aspect-oriented software development (AOSD, [2]) and USE. We point out how the development of mobile telecommunication systems can benefit from the deployment of AOSD and the provisioning for USE.

The paper is organized as follows: Section 2 illustrates our approach to DSA, addressing modularity and variation points, aspect-oriented programming, late binding and reflection. It also gives an overview of our research platform. Section 3 demonstrates DSA applied in the context of runtime system integration and extension. Section 4 outlines further opportunities for DSA. Section 5 concludes the paper.

2 Dynamic Service Adaptation

The concept of adaptability is closely related to that of modularity and variation points. The modularization of a system can improve its flexibility and comprehensibility, and with that can also shorten its development time. Variation points provide us a way to explicitly designate module boundaries in a system's design where changes are expected to happen. The introduction of variation points and with it the separation and composition of common and variable system aspects can provide for flexibility.

The majority of recently built systems are based on object-oriented technologies. Here, classes and instances are employed as both modularity constructs and units of change. Besides other important properties, most aspect-oriented programming (AOP) technologies provide a new, finer grained, modularity construct that allows us to represent crosscutting concerns, down to the methods of individual instances.

Since many changes happen after a system's initial deployment, they need to be addressed very late in its lifecycle. To avoid system downtimes, many of the corrective actions covering these changes need to be performed on demand at runtime. We consider reflective architectures and late binding to be key elements of a DSA platform addressing these requirements. In our approach to DSA, we use the aspect modularity construct to adequately represent units of change. Computational reflection, dynamic AOP, and late binding will allow us to adapt service and service

platforms as late as possible, preferably without system downtimes and with that the disruption of service [11].

In the following subsections we give an overview of modularity, variation points, AOP, reflection, and late binding. The last subsection outlines our research platform for DSA and runtime system extensions.

2.1 Modularity, Variation Points, Objects, and Aspects

One approach to manage complexity is modularity. Here, we are trying to improve the comprehensibility and flexibility of a system by decomposing a complex system into smaller, less complex subsystems and then recomposing these subsystems in a principled way. Modules help to hide from each other complex design decisions or design decisions which are more likely to change [22]. Variation points, or hotspots [23], designate module boundaries in a system's design where changes are expected to happen without the need to explicitly name all of them. With variation points we improve flexibility in the context of change through the separation and composition of common and variable aspects of our system.

Variations and variation points depend to a large extent on the underlying modularity mechanism provided by the programming platform a system is built on. Most modern software systems were built using object-oriented technologies where the modularity constructs, and with that the units of change, are that of classes and instances. Here, classes capture the properties of their instances. Although this level of granularity is sufficient in some cases, a more fine-grained approach to modularity is desirable to permit the change of even smaller semantic units such as method implementations.

In object-oriented systems there is code that, even though it implements one particular concern, is spread around (scattered) over many or even almost all modules, crosscutting various other modules implementing other concerns as well, instead of being confined to one or a small number of modules. Because of its non-explicit structure, such crosscutting code is hard to comprehend and difficult to change. The consistency of changes is both hard to verify and to enforce. Object-orientation and its class modularity construct, while proven to be appropriate for many modeling scenarios, cannot be of help in implementing other concerns in a modularized way. Also, while traditional modules such as classes and instances might support the proper structuring of the initial system, subsequent changes to this system could crosscut these module boundaries to affect more than one location.

Based on the assumption that crosscutting is inherent to complex software systems, AOP ([6, 16]) as a new software technology addresses the issues of separation of concerns (SOC, [5, 12]). For that, AOP introduces orthogonal units of modularity to capture crosscutting structures explicitly. Such structures are called aspects and can be found in a software system's requirements, its design, as well as in its implementation. AOP builds on existing technologies but provides additional mechanisms that make it possible to affect a system's implementation in a crosscutting way [4]. Aspects are units of modularity that represent implementations of crosscutting concerns. Aspects associate code fragments (code to be executed when a join point is encountered) with join points (well-defined points in the execution of code) by the use of advice constructs. A collection of related join points descriptors, to be addressed by an advice, is called a pointcut. Join point descriptors denote targets

for the weaving process to apply changes to the underlying computational base system as stated in the advice constructs.

Aspects and their advice are integrated into the base system during an activity called weaving. Weaving in general can be performed at almost any point in time in a software system's lifecycle. Most of today's AOP technologies limit themselves to either compile-time, load-time, or runtime. AspectJ [15] and HyperJ [25] are examples for compile-time weaving. In AspectJ for example, the weaver parses an AspectJ program, transforms the AspectJ abstract syntax tree (AST) into a valid Java [8] AST, and then generates Java byte code for a standard Java virtual machine. JMangler [18] performs load-time transformation of Java class files. AspectS [9] employs run-time weaving to transform the base system according to the aspects involved. The woven code is based on method wrappers [3], reflection [20, 24] and meta-programming [17].

As of today there are several approaches supporting aspect-oriented concepts, ranging from domain-specific aspect languages such as RG [21] or D [19] to general-purpose aspect languages like AspectJ or AspectS. Many of these languages allow us to express crosscutting concerns, down to the level of individual instances, methods and variables. Like objects in object-oriented software development, aspects may appear at all stages of the software development lifecycle. Illustrative examples of aspects that can be commonly observed are architectural or design constraints, features, and systemic properties or behaviors.

2.2 Late Binding and Computational Reflection

Software development is still hard. During the software development lifecycle we quite frequently find out something we wished we had known from the very beginning of the project [14]. While there is always a chance that some of the requirements were not sufficiently understood to adequately address them in the software system, many changes happen after a system's initial deployment, and so are impossible to predict and dealt with right from the beginning. On the contrary, such changes must be addressed very late, after deployment. System downtimes can be minimized if most corrective measures can be applied at runtime. To address this requirement, we consider late binding and reflective architectures to be key elements of a DSA platform.

Late binding is a mechanism to defer decisions to a later point in time which allows us to avoid too early commitments to design decisions, especially decisions regarding variation points, we might or will not be able to maintain. Whereas early binding requires us to provide abstractions addressing possible change at a very early point in time, late binding helps us to avoid such premature abstractions. Extreme late binding allows these decisions to be made as late as possible, at runtime.

Systems with reflective architectures incorporate structures representing aspects of themselves [20]. The aggregate of these structures is called the system's self representation which allows the system to both observe its own computation as well as influence or change it. The activity of observing oneself is called introspection, the activity of changing oneself intercession. For service adaptation, introspection will allow us to observe computational properties of a deployed set of services as well as the computational environment they are running in. Intercession can be based on our observations and result in the alteration of the service.

2.3 Adaptation Platform

While most of today's adaptation mechanisms focus on content and a few on communication, almost none considers the adaptation of service logic itself. Because of that, our research on DSA is directed towards behavior adaptation at runtime. To adequately address change, services and service platforms need to be adaptable, as late as possible, when changes actually require adaptation to happen. Making changes effective dynamically at runtime will offer the benefit of avoiding system downtimes, and with that the disruption of service.

Our DSA research platform is based on a layered system architecture. Squeak/Smalltalk serves us a very dynamic object-oriented multimedia scripting environment [7, 13]. AspectS extends the Squeak/Smalltalk environment to allow for experimental aspect-oriented system development. PerspectiveS builds on AspectS to allow for dynamic behavior layering in the Squeak environment.

Squeak/Smalltalk's properties that are important to our research on DSA are its extensive reflection support covering both introspection and intercession, its powerful metaobject protocol [17] that gives us full access to the computational properties of our environment, and its support for very late binding to defer binding decisions until the point when they actually need to be made. The idea of metaobject protocols is that one can and should open languages up to allow users to adjust the design and implementation to make the language or environment to suit their particular needs.

AspectS provides a platform for the exploration of aspect-oriented software composition in the context of dynamic systems [9]. It allows for convenient meta-level programming, addressing the tangled code phenomenon by providing aspect modules. AspectS shows great flexibility by not relying on source or bytecode code transformations. Instead, it makes use of metaobject composition. In contrast to most other approaches to AOP that only focus on class-level aspects, AspectS allows for instance-level aspect and with that for modularization of behavior that crosscuts individual instances.

PerspectiveS coordinates the activation of a set of aspects, and so lets us to decorate a system with context-dependent behavior, without requiring developers of the base system to be aware of that [10]. PerspectiveS enables greater separation of concerns of a base system from its context-dependent behavior. Here, base systems can be freed from providing behavior that explicitly takes action in response to context changes not known at neither development- nor deployment-time. PerspectiveS facilitates basic role modeling by dynamic composition of multiple roles without the loss of object identity. Roles can be added or removed on-demand, with each role bringing in its own set of state and behavior.

All of these layers allow us to both implement our basic service logic as well as to adapt this service logic to additional requirements and unforeseen circumstances if necessary. Due to the dynamic nature of our research platform, adaptation activities can be carried out on an on-demand basis, during runtime, while our services are already deployed and activated [11].

In the following section, we use a scenario of runtime system extension to illustrate the application of our DSA platform.

3 Runtime System Extensions

Next generation mobile communication systems will give third-party service providers more opportunities to offer their service on a variety of open service platforms. Since there will be several such platforms, services are likely to not match all of them in the same way: While some services and some platforms are perfect matches, in many cases there is some work to be done to integrate them adequately to ensure an pleasant service experience. As already stated, it is not possible to identify and apply all of these changes upfront, right from the beginning. Most of them are to become effective after the initial deployment a service or its service platform. And preferably all of them should be applied without noticeable disruption of service.

Our work on DSA not only covers the design and implementation of an adaptation platform, but also includes the illustration of our approach by describing candidate scenarios of DSA. In [11] we explain the application of our DSA platform to integrate a third-party component, the Fauré personal digital assistant (PDA) [1], and the value of DSA by discussing four scenarios: The introduction of additional safeguards let us correct the wrong assumption of the Fauré component provider that this PDA component would be operated standalone and terminating it requires quitting the underlying platform, too. The enforcement of style guide elements allowed us to change the original appearance of the user interface (UI) to conform to the requirements of a particular style guide – be it because of a difference in the style offered by the component and the style required by the platform operator, or because the style guide of the platform operator was changed itself and all existing components conforming to the previous style guide now have to conform to the new one. Late UI branding let us decorate suitable UI elements with brand names, logos, or advertisements. In the category of upgrades, updates, and fixes we resolved an issue with the rendering engine of our platform we discovered while carrying out our late UI branding adaptation.

In the following we will show how to take advantage of DSA to extend the Fauré PDA component with another service application, and to instrument the newly integrated application with a notification mechanism to indicate proper usage indication events.

3.1 Our Base Application

We will use the Fauré PDA [1] as the service application to be extended dynamically at runtime. Fauré, an open source PDA implementation for Squeak designed to run on a handheld device, runs on top of our DSA platform, most likely on a mobile terminal.



Fig. 1. Fauré Welcome Screen

When we launch our PDA application, its welcome screen shows summarized list of our things to do and our personal schedule (Fig. 1). Via the view menu, we can reach to our full contacts database, all of our social events and things to do, a little sketch pad, a piano-like music instrument, and a demonstration of the 3D rendering facilities of Squeak.

3.2 Tetris

The Fauré PDA also provides a game called Same Game, originally written by Eiji Fukumoto for UNIX and X. The object of SameGame is to maximize the score by removing tiles from the board. Tiles are selected and removed by clicking on a tile that has at least one adjoining tile of the same.

But what if most of our customers would like to play another more popular game, a game like Tetris? Tetris was originally developed by Alexey Pazhitnov on an Electronica 60. In Tetris, regularly-shaped blocks appear at the top of the screen and advance steadily down a fine grid. These blocks can be spun to make them fit into point-scoring rows. As levels get completed, Tetris is getting faster what makes it harder to spin and fit blocks together to complete the rows.

3.3 Tetris Integration

After searching for an implementation of Tetris, we find one the runs in our execution environment (Fig. 2). Unfortunately, that implementation does not fit into our PDA: The UI element representing the game is too large because its height exceeds the height made available for user applications by the PDA. Also, the game control buttons that allow us to rotate and drop Tetris pieces are in a location that would cause us to waste even more screen real estate we cannot afford.

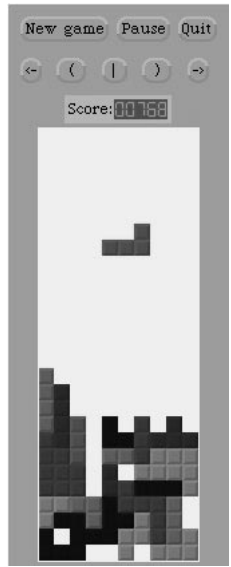


Fig. 2. Tetris

A common approach to make the new Tetris game fit into the PDA environment would be to obtain its source code, change this source code, and completely rebuild the game application. Another way to make Tetris conform to our requirements is to provide an additional piece of software that instructs our runtime environment on how to transform this game to become deployable within our provisioning environment.

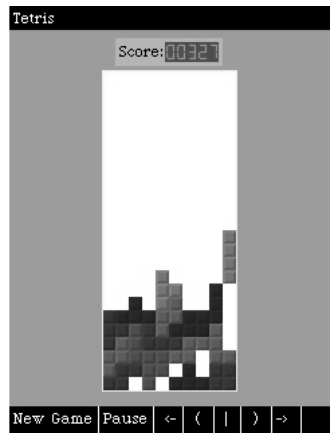


Fig. 3. Adapted and Integrated Tetris

Fig. 3 shows the same Tetris applications previously discussed after its transformation and integration into the PDA. One can see how its size was changed to meet the constraints imposed by the PDA. Also, all game control buttons previously

found on top of the game area are now arranged in the bottom row of the PDA UI where one would have placed them in the first place if the game would have been designed to run in the PDA from the beginning.



Fig. 4. Fauré View Menu with Tetris Menu Entry

Making Tetris fit is not enough to claim its integration is done. It needs to be accessible by the user, too. For that we have to extend the launch menu of our PDA by providing an entry that will launch Tetris if selected. Fig. 4 shows the extended menu, with our new Tetris entry last in the list.

3.4 Usage Indication and Metering

Merely providing new applications and services to our customers might not be sufficient from a business' point of view. Providing services implies most of the time some form of compensation, either directly or indirectly. Compensation is typically based on service level agreements (SLAs) about quantitative information about the usage of a service. Since most of the time third-party software components are not developed to target specific SLAs and also because SLAs can change as often as possible, it is not of benefit to commit to specific usage indications too early in the service lifecycle.

DSA allows us to instrument our applications and services to provide usage indication information, not even after their development, but also after their deployment, as late as at runtime.

Fig. 5 shows a usage indication trace of Tetris, where each start of a new game is reported to usage collection mechanisms which can act as an input feed to a rating and billing engine. This usage indication record generation was introduced by one of our adaptation modules that instrument the original Tetris component.



Fig. 5. Posted Tetris Usage Indication Records

The following listings illustrate how this adaptation was achieved. The first listing shows the method that gets invoked every time a customer presses the ‘New Game’ button (`TetrisBoard>>newGame`): Tetris starts over with a new game.

TetrisBoard>>newGame

```
self removeAllMorphs.
gameOver _ paused _ false.
delay _ 500.
currentBlock _ nil.
self score: 0.
```

In the next listing we can see code that belongs to our adaptation module (`FdsaTetrisUsageAspect`) and is responsible for instrumenting the `newGame` Method in such a way that every time (except for the first) it gets invoked, a usage indication record will be posted to the responsible entity (in this simplified case the system transcript, Smalltalk’s console).

FdsaTetrisUsageAspect>>adviseTetrisBoardNewGame

```
^ AsBeforeAfterAdvice
  qualifier: (AsAdviceQualifier
    attributes: { #receiverClassSpecific. })
  pointcut: [OrderedCollection
    with: (AsJoinPointDescriptor
      targetClass: TetrisBoard
      targetSelector: #newGame)]
  afterBlock: [:rcvr :args :aspect :client :return |
    thisContext baseSender baseSender selector
      ~~ #initialize "the first game is for free"
      ifTrue: [self postTetrisUsage]]
```

The convenience method `postTetrisUsage` is implemented as follows:

FdsaTetrisUsageAspect>>postTetrisUsage

```
Transcript
  cr; show: '<UsageIndicationRecord User="' ,
```

```
self userIdentifier printString,  
  '" Application="Tetris" Date="',  
  Date today printString, '" Time="',  
  Time now printString, '" Usage="NewGame">'.
```

Our deployed PDA service will be accompanied by the Tetris component to be integrated and the adaptation modules necessary to do so. The adaptation module shown above is only responsible for dynamic usage indication record generation, the adaptation module required to integrate Tetris into the PDA service is not shown in this paper.

4 Further Opportunities

There are quite a few opportunities for DSA and runtime system extensions. The following subsections will illustrate how personalization, application-level security, pre-standard releases, and addressing regulatory requirements can benefit from DSA.

4.1 Personalization

Personalization is regarded to be one of the most compelling features for mobile communications systems B3G by supporting users in selecting the best services from the rapidly increasing diversity of mobile services, and adjusting selected services to their individual needs. Service personalization promises to foster and improve the relationship between service providers, mobile operators and customers. Also, it is expected to promote the adoption of increasingly complex services.

Service personalization can basically be approached from two points of view. On the one hand, there is the user perspective where user models are developed and expressed through user profiles and user preferences within the respective system of client devices, services, and applications. We consider context awareness to be an integral part of this user-centered position.

On the other hand, there is the system side where we need to consider how personalization features are implemented and how the personalization of mobile applications effects actual system execution and runtime behavior. An example is the impact of changes in a user profile on the service delivery in a given situation, for instance taking the change of a user's geographical location into consideration.

Service personalization is not limited to data (such as selective content delivery) or the user interface only, but will also involve DSA with changes to behavior (service logic) and interaction (service signaling and communication). Such service adaptation allows mobile systems to react to changes in the environment, which are inherent in their nature to users roaming in a federated world-wide service space. For instance, a personalized software application may be downloaded by users, based on their personal preferences or current environments. To implement this, the appropriate user aspects have to be merged for a personalized service.

DSA also supports the creation of services being capable of dynamic personalization. For example, adaptation of service behavior can be made possible

through extensions to the service provisioning infrastructure that allows the selection of units of modularity to be adjusted.

4.2 Application-Level Security

One of the main acceptance criteria for new communication and collaboration services is an adequate management of privacy. We need to ensure all privacy policies and security constraints to be enforced consistently across the whole system. Furthermore, in an open environment, we need to assure that our security restrictions and privacy policies not only affect components currently installed and running, but also the ones that will be installed in the future.

To ensure that, we need a mechanism that continuously observes the runtime platform and adjusts to the requirements all newly added components in a consistent manner. DSA can play an important role in providing such a mechanism.

4.3 Pre-standard Releases

Very often, standardization processes take a long time, and most of the time longer than expected. While shipping standard conformant products is essential for solutions that have to be integrated with a heterogeneous environment, time to market is most of the time more critical to the success of a business than standard conformance.

DSA allows for both, early product releases and standard conformance. If early releases of a standard become reasonably stable, affected component can be released at that time. Once the final release of the standard becomes available, all affected and already deployed components can be updated to conform to the available standard. Advanced product planning will be possible through the application of DSA in later phases of the lifecycle of a product.

4.4 Regulatory Requirements

The same said about pre-standard releases holds for regulatory requirements to be met. Whenever changes of laws or other regulations affect products and systems already released and deployed, such products and systems need to be adjusted. This process can become very cost intensive if carried out the traditional way by building a completely new system, taking down the old systems and bringing up the new ones, possibly with the consequence of service outages and all economical consequences involved.

DSA allows us to upgrade deployed and running systems, at runtime, without the need to disrupt any service provided. Delta modules can provide the additional or changed functionality needed to meet new requirements, and the DSA infrastructure makes these modules effective without service disruption if possible.

5 Summary

We expect next generation mobile communication systems to be more open to third-party service providers, yielding a rich and flexible service landscape. With that, such systems will be more complex than ever before. Different parts of the system will evolve at a different pace. Service offerings continuously come and go. And because change is rather the norm than the exception, service platforms need to prepare for it. Instead of relying on premature abstractions, other mechanisms are required to allow for system adaptations to be performed – when they are needed, on-demand. To ensure a pleasant service experience and to avoid system downtimes and disruptions of service as much as possible, necessary adaptations should preferably be carried out during runtime. In this paper we show what we believe is necessary to dynamically adapt services by giving an overview of our approach, our adaptation platform, and by showing how to apply these concepts and technologies to integrate and extend services at runtime. While in the past most of the adaptation strategies are based on redundancy and failovers, this is no longer possible anymore in a world of small mobile devices. A new approach is required to deal with change. DSA is ours.

Acknowledgements. We would like to thank Matthias Wagner, Stefan Hanenberg, Andreas Raab, Wolfgang Kellerer, Anthony Tarlano, and Christian Prehofer for their contributions.

References

1. Allen, R.: *Faure*. <http://russell-allen.com/squeak/faure/>.
2. Aspect-Oriented Software Development homepage (<http://www.aosd.net/>).
3. Brant, J.; Foote, B.; Johnson, R.; Roberts, D.: *Wrappers to the Rescue*. In: Proceedings of the 1998 European Conference on Object-Oriented Programming (ECOOP), pp. 396–417, Brussels, Belgium, 1998.
4. Elrad, T.; Aksit, M.; Kiczales, G.; Lieberherr, K.; Osher, H.: Discussing Aspects of AOP. In: *Communications of the ACM*, Vol. 44, No. 10, pp. 33–38, October 2001.
5. Ernst, E.: *Separation of Concerns*. In: Proceedings of the AOSD 2003 Workshop on Software-Engineering Properties of Languages for Aspect Technologies (SPLAT), Boston, MA, USA, March 2003.
6. Filman, R.E., Friedman, D.P.: *Aspect-Oriented Programming is Quantification and Obliviousness*. In: Proceedings of the ECOOP 2001 Workshop on Advanced Separation of Concerns, Budapest, Hungary, June 2001.
7. Goldberg, A.; Robson, D.: *Smalltalk-80: The Language and Its Implementation*. Addison-Wesley, 1983.
8. Gosling, J.; Joy, B.; Steele, G.; Bracha, G.: *The Java Language Specification (Second Edition)*. Addison-Wesley, 2000.
9. Hirschfeld, R.: *Aspects – Aspect-Oriented Programming with Squeak*. In: M. Aksit, M. Mezini, R. Unland, editors, *Objects, Components, Architectures, Services, and Applications for a Networked World*, LNCS 2591, pp. 216–232, Springer, 2003.
10. Hirschfeld, R.; Wagner, M.: *PerspectiveS – AspectS with Context*. In: Proceedings of the OOPSLA 2002 Workshop on Engineering Context-Aware Object-Oriented Systems and Environments (ECOOSE), Seattle, WA, USA, 2002.

11. Hirschfeld, R.: *Dynamic Service Adaptation*. DoCoMo Euro-Labs Technical Report, ITR-FNL-023, Munich, April 2003.
12. Hürsch, W.L.; Lopes, C.V.: *Separation of Concerns*. College of Computer Science, Northeastern University, Boston, USA, February 1995.
13. Ingalls, D.; Kaehler, T.; Maloney, J.; Wallace, S.; Kay, A.: *Back to the Future: The Story of Squeak, a Practical Smalltalk Written in Itself*. In: Proceedings of the 1997 Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA), pp. 318–326, Atlanta, GA, USA, October 1997.
14. Kay, A.: *Is “Software Engineering” an Oxymoron?* Viewpoints Research Institute, 2002.
15. Kiczales, G.; Hilsdale, E.; Hugunin, J.; Kersten, M.; Palm, J.; Griswold, W. G.: *An Overview of AspectJ*. In: Proceedings of the 2001 European Conference on Object-Oriented Programming (ECOOP), pp. 327–355, Budapest, Hungary, 2001.
16. Kiczales, G.; Lamping, J.; Mendhekar, A.; Maeda, Ch.; Lopes, C. V.; Loingtier, J.-M.; Irwin, J.: *Aspect-Oriented Programming*. In: Proceedings of the 1997 European Conference on Object-Oriented Programming (ECOOP), pp. 220–242, Jyväskylä, Finland, 1997.
17. Kiczales, G.; des Rivieres, J.; Bobrow, D.: *The Art of the Metaobject Protocol*. Addison-Wesley, 1991.
18. Kniesel, G.; Costanza, P.; Austermann, M.: *JMangler – A Framework for Load-Time Transformation of Java Class Files*. In: Proceedings of the Workshop on Source Code Analysis and Manipulation (SCAM). Florence, Italy, November 2001.
19. Lopes, C. V.: *D: A Language Framework for Distributed Programming*. Dissertation. College of Computer Science, Northeastern University, Boston, USA, 1997.
20. Maes, P.: *Concepts and Experiments in Computational Reflection*. In: Proceedings of the 1987 Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), pp. 147–155, Orlando, FL, USA, 1987.
21. Mendhekar, A.; Kiczales, G.; Lamping, J.: *RG: A Case-Study for Aspect-Oriented Programming*. Xerox PARC. Technical Report SPL97-009 P9710044. February 1997.
22. Parnas, D.L.: On the Criteria To Be Used in Decomposing Systems into Modules. In: *Communications of the ACM*, Vol. 15, No. 12, pp. 1053–1058, December 1972.
23. Pree W.: *Design Patterns for Object-Oriented Software Development*. Addison-Wesley, 1994.
24. Rivard, F.: *Smalltalk: A Reflective Language*. In: Proceedings of Reflection 1996.
25. Tarr, P.; Ossher, H.; Harrison, W.; Sutton Jr., S.M.: *N Degrees of Separation: Multi-Dimensional Separation of Concerns*. In: Proceedings of the International Conference on Software Engineering (ICSE), pp. 107–119, Los Angeles, CA, USA, May 1999.

Flood Filtering and Route Selection for Energy-Efficient On-Demand Routing in Wireless Ad Hoc Networks

Tran Minh Trung and Seong-Lyun Kim

Radio Resource Management & Optimization Laboratory

School of Engineering

Information Communication University (ICU)

Yusong P.O. Box 77, Taejon 305-600, Korea

{trungtm, slkim}@icu.ac.kr

Abstract. This paper presents a new routing algorithm for maximizing the lifetime of a wireless ad hoc network. Our approach is to reduce the energy consumption at routing discovery phase and to establish suitable routing paths with respect to energy-saving. At the routing discovery phase, we use a dynamic flood filtering algorithm to eliminate the nodes that are predicted not to participate in the final routing paths. The prediction rules are based on energy requirement by each connection, energy availability at each node, and neighbor nodes' status. Finally, at the destination node, by using an energy-efficient route selection algorithm, the selected routing path would have small energy consumption with an ample residual energy capacity.

1 Introduction

The wireless ad hoc network is composed of mobile, wireless nodes; its availability depends on status of each node. For that, the failure of any node can divide the network into multiple parts. In particular, if a node runs out of energy, the probability of network partitioning will increase. Since every mobile node has a limited power supply, the energy depletion becomes one of the main threats to the lifetime of the ad hoc network. To cope with the issue, recently proposed solutions have tried to find energy-efficient routing algorithms, with hope that the power consumption can be distributed evenly among the nodes (see [1–8] and references therein).

In this paper, we approach the issue from a different angle. *On-demand routing protocols* such as AODV [9] and DSR [10] are most widely accepted algorithms by wireless ad hoc networks. However, when applying those on-demand routing protocols, the flooding of control messages such as RREQ (route request) packets at the *routing discovery phase* will be out of control. Since the flooding is one of the most energy-intensive operations, the uncontrolled flooding will lead to unnecessary energy consumption at nodes, resulting in serious redundancy, contention and collision (known as *broadcast storm* [11]). Our focus is how to

reduce the energy consumption in the routing discovery phase, while the selected path still has the reasonable energy capacity. Regarding the broadcast storm, there have been some previous researches [11-15] for reducing redundancy of the simple flooding, which can be categorized as follows:

- *Probabilistic-based scheme* [11, 12, 14]: Whenever a node receives an RREQ message for the first time, it will re-broadcast the message with a probability P . This scheme becomes worse (just like the simple flooding) when we want to increase the reach-ability performance by increasing P .
- *Counter-based scheme* [11, 14]: When the medium is busy and the queued messages are many, there is a chance that a node has received the same RREQ message many times before the node starts re-broadcasting it. As a remedy, the node keeps track the number of times the same RREQ message has been received. If it is larger than a threshold, re-broadcasting is inhibited.
- *Distance-based scheme* [13]: Having understood the relationship between the distance and the power, we can even directly replace the role of distances by signal strengths by establishing a signal-strength threshold, and then make re-broadcast decision. The signal strength information was also used to facilitate routing.
- *Location-based scheme* [11, 14]: It uses GPS to get precise location of each node. The location information is used to facilitate the route discovery process.
- *Cluster-based scheme* [11, 13, 15]: This approach is based on a graph-theoretic modeling. All nodes are partitioned into clusters. Each node can communicate with other nodes through a host node, called gateway.

From lifetime maximization point of view, some nodes may not be selected for the final route. For example, a node that has a low residual battery capacity compared to energy requirement by the incoming connections, may be better to be filtered out in the route discovery phase. All of above mentioned flooding schemes, however, did not consider the energy requirement of incoming connections, the residual energy of each node and the impasse zone (non-destination) that routing control messages should not be forwarded to. Consequently, they still have to waste an unnecessary amount of energy for forwarding redundancy packets. Also, we think some of above mentioned algorithms are too complicated for being implemented in the real world. Our goal in this paper is to remedy such drawbacks of the existing flood-filtering algorithms while having the practical applicability.

In this paper, we propose so-called the *dynamic flood filtering* that considers four aspects: (i) node residual energy capacity (ii) node's affordability for incoming connections (iii) neighbor nodes' link status, and (iv) practical applicability of the flood filtering. Our idea is to predict whether a node has enough capability for participating in the final routing path or not, based on its available energy capacity and its neighbor's link status. If a node does not have favorable conditions, it will not have to take part in flooding process. Our aim is to apply our idea to AODV or DSR, achieving energy-efficiency while keeping the on-demand

behavior. One minor contribution of this paper is in the new route selection algorithm (after the flooding is finished) that selects a suitable routing path with respect to energy-saving. To see the effectiveness of our idea, we compare it with some other routing algorithms, such as *minimum hop* (MH) and *min-max battery cost routing* (MMBCR) [4], in which AODV route discovery was combined. The remainder of this paper is organized as follows. Section II describes our proposed solution. Our simulation results and evaluation are presented in Sections III. Section IV concludes the paper with remarks on ongoing research.

2 Proposed Solutions

2.1 Basic Ideas

The routing protocol that we choose to apply our proposed solutions is ADOV. In the original on-demand routing protocol, when receiving an RREQ, a node that is unable to be a destination and does not have a routing path to the destination in the routing table, will re-broadcast it to other nodes. This action will be repeated until the RREQ reaches the destination or the node that eventually has a routing path to the destination. Since only few nodes can take part in the selected routing path while many others have to only re-broadcast the RREQ packet, the routing discovery performs many redundant re-broadcast. To solve this problem, we propose a dynamic flood filtering algorithm. We consider three cases in which we can predict whether a node will belong to the final selected routing path or not. From that, we can eliminate unnecessary nodes from the routing discovery phase:

- *Weak node*: A node that does not have enough energy for serving the incoming connection. For example, a source node wants to send a big amount of data, where the minimum energy requirement for sending and receiving this amount of data is 100 joules. However, when the relay node has only 20 joules left, so it drains out energy after a short time. Consequently, the connection will be disconnected and the routing discovery phase will be activated again. That makes unnecessary energy consumption. Figure 1 shows that when the energy requirement of the connection from source (A) to destination (F) is 100 joules, the selected path will be (A, D, E, F). The path (A, B, C, F) was not selected because it contains node B that does not have enough energy for serving the required connection.
- *Congested node*: A node has enough residual energy capacity but it coincidentally belongs to some other active routing paths. So its energy will be drained out very fast. If the energy for serving the current connections and the coming connections is not enough, then the node would be better reject incoming connections. Figure 2 shows that node X has enough energy for serving a connection that requires energy smaller than 120 joules, but it coincidentally belongs to two active routing paths: (B, X, E) and (D, X, C). So as time goes by, the residual energy of node X will go down very fast and the node cannot guarantee enough energy for serving the incoming connections.

- *Black zone*: Except the node that it receives the RREQ from, a node may have only one hop neighborhood, all of which as well as itself are not the destination. Figure 3 shows that an RREQ message is broadcast to an impasse (or a black zone), where none of the nodes B, C, D is the destination. Except node A, node B has only nodes C and D as its neighbors.

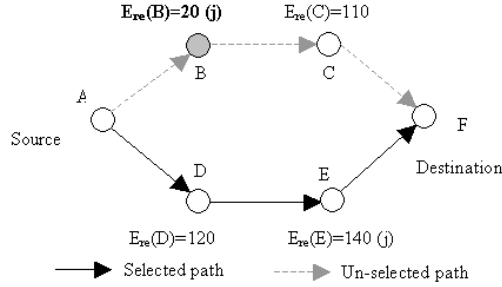


Fig. 1. The node B should be eliminated from routing discovery phase.

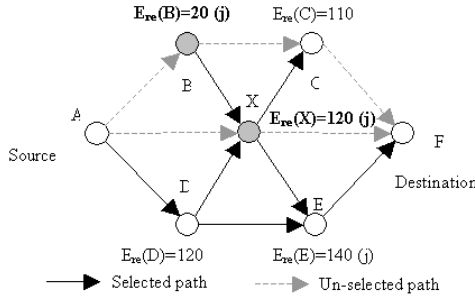


Fig. 2. Node B, X should be eliminated from routing discovery phase.

2.2 Dynamic Flood Filtering Algorithm

In order to incorporate these three aspects into AODV, we consider to modify HELLO and RREQ messages of AODV as follows:

- *HELLO message*: In original AODV, each forwarding node should keep track of its continued connectivity to active neighbor hops (i.e., which next hops or precursors the current node has sent/received packets to or from). This is done by using the HELLO messages that are exchanged periodically among

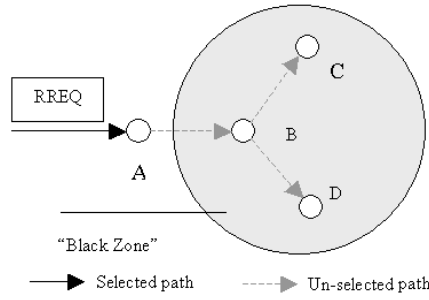


Fig. 3. Node B, C and D should be eliminated from routing discovery phase.

Table 1. The status table of node B of Figure 3. In this table, the LINKS_STATUS value of node A is 2, which means that B has connection up to at least 2 hops via node A, because A has other neighbor nodes other than B.

ID	Residual Energy	Links Status
A	20(j)	2
B	100(j)	0
C	140(j)	1
D	90(j)	1

the nodes. In our flood filtering, we suggest to use two more fields in the HELLO message, for constructing/updating the *neighborhood status table* (e.g. Table 1): (i) RESIDUAL_ENERGY field contains information about the residual energy capacity of a node. (ii) LINK_STATUS field contains information about the link status of neighborhood nodes. We use three values to indicate the link state of a node. The record that contains “0” value belongs to the node that owns this status table. The value being greater than “0” indicates that the corresponding neighbor node has connection upto one or more than one hops away. It is notable that our modification of the HELLO message costs exchanging few more extra bits among the nodes.

- *RREQ message*: Whenever a source node initiates a routing process for a connection, the information about *incoming energy request* ($E_{future_request}$) of this connection will be piggybacked into ENERGY_REQUEST field of the RREQ header. The value $E_{future_request}$ is calculated as follows:

$$E_{future_request} = N_{packets} \times (E_{tx} + E_{tr}) \quad (1)$$

where E_{tx} and E_{tr} denote energy consumption for transmitting and receiving one packet, respectively. The number $N_{packets}$ denotes the total number of packets that will be transmitted over the connection. Also, we suggest

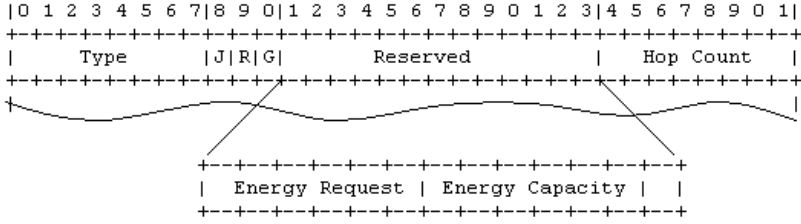


Fig. 4. Modified AODV RREQ Header. 13 bits are reserved for storing ENERGY_REQUEST and ENERGY_CAPACITY information [9].

using one more field in RREQ header; ENERGY_CAPACITY field stores the information about *energy capacity* ($E_{capacity}$) of a routing path.

Since the energy capacity of each possible routing path is equal to the *available energy capacity* ($E_{available(i)}^t$) of the weakest node on the path, it is calculated as follows:

$$E_{capacity} = \text{Min}_i[E_{available(i)}^t] \quad (2)$$

where $E_{available(i)}^t$ is the predicted available energy capacity of node i at time t and the time t denotes the instance that the RREQ packet passes through node i . This available energy capacity is calculated based on its residual energy capacity and the energy requirement by the current active routing paths. Let $E_{current_request(j)}^t$ be the energy request by active routing path j at time t then:

$$E_{available(i)}^t = E_{residual(i)}^t - \sum_j E_{current_request(j)}^t \quad (3)$$

Each time the RREQ message passes by a node, the value $E_{capacity}$ in ENERGY_CAPACITY field will be compared with $E_{available(i)}^t$ of that node. If it is greater than $E_{available(i)}^t$ then ENERGY_CAPACITY field will be updated with $E_{available(i)}^t$. By performing this method, the ENERGY_CAPACITY field always contains the minimum available energy capacity of the nodes on the path that the RREQ packet has traversed. The modified format of RREQ message header is shown in Figure 4, in which we use 6 reserved bits for storing energy request information and the other 6 reserved bits for storing energy capacity information.

In the original AODV routing protocol, at routing discovery phase, when a node receives an RREQ, it determines whether it has received RREQ:s from the same originator IP address, during at least the last PATH_DISCOVERY_TIME. If such an RREQ has been received, the node silently discards the newly received RREQ. Else, if this node is not the destination or it does not have an active routing path to the destination, it will re-broadcast the RREQ message to all neighboring nodes. In order to eliminate unnecessary nodes from the routing discovery phase, we add two checking processes into the *processing and forwarding route request step* as follows:

- *Weak and congested node*: Whenever a node receives an RREQ message, it will compare its *available energy capacity*, $E_{available(i)}^t$ with the *energy request*, $E_{future_request}$ of the incoming connection. If the former is smaller, then this node will silently discard the RREQ packet. Figure 5 shows that, node (5) discards an RREQ message because its available energy capacity is smaller than the energy request. While node (10) has enough residual energy capacity but it is on-service for two ongoing connections: (11, 10, 12) and (5, 10, 13) at the same time. As a result, its available energy may not be enough for the incoming connection; it will discard the RREQ packet. In the case that all the neighbor nodes do not have enough available energy capacity, the RREQ will broadcast with $E_{future_request}$ value equaling to the maximum available energy capacity of the neighbor nodes.
- *Black zone*: In order to prevent the RREQ packet from being broadcast to a black zone (Figure 3), each node receiving an RREQ will check its neighbors' link status. If all of the neighbor nodes, except the node that the RREQ is sent from, are not the destination and have LINK.STATUS value smaller than '2', it will discard the RREQ packet. In Figure 5, we can see a black zone starting at node (1) so that the node will discard all RREQ packets. This case is the same as those of nodes (2) and (16).

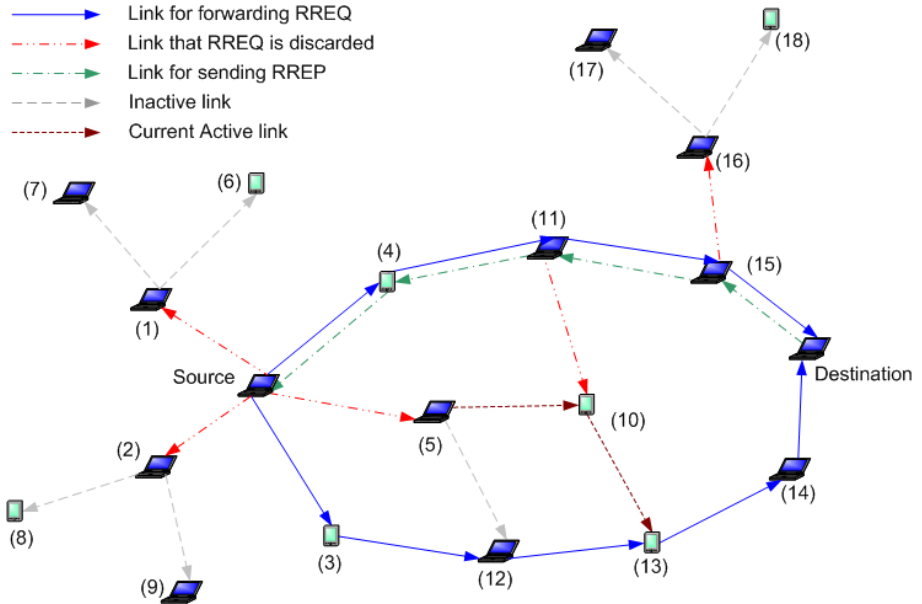


Fig. 5. A sample RREQ flooding with 7 forward nodes and 11 non-forwarding nodes.

2.3 Energy-Saving Route Selection Algorithm

During a receiving period, each received RREQ packets will be saved in a lexicographic order with two entries: (HOP_COUNT, ENERGY_CAPACITY). After the period, the RREP (route reply) will be generated, with respect to the first RREQ packet in this lexicographic order. By this way, we hope that the selected routing path would have smallest energy consumption for transmitting/receiving data among the possible routing paths. Due to the filtering, the selected path has reasonable energy capacity for serving the incoming connection. Figure 5 illustrates our flooding process. Instead of 16 nodes re-broadcasting RREQ packets, there are only 7 nodes performing RREQ re-broadcast. The possible routing paths at the destination are: (S, 4, 11, 15, D) with 4 hops and (S, 3, 12, 13, 14, D) with 5 hops. The final selected path is (S, 4, 11, 15, D) with smaller hops (energy consumption) for transmitting and receiving data.

3 Simulation Results and Evaluations

To get the insights on the proposed solutions, we compare it with two other routing algorithms; one is the minimum hop (MH) and the other is MMBCR [4]. All of them are combined with the RREQ flooding of AODV. Each algorithm is implemented by the NS2 tool [16]. In our simulation model, 40 mobile nodes are generated randomly in an area of 500m×500m. The moving speed of each node is 10m/s. A number of connections with different levels of energy requirement are established during 900 seconds. In this simulation we choose the energy requirement of each generated connection is equal with $\frac{3}{4}$ of the residual energy capacity of the source node. For instance, if at simulation time t , a source node has residual energy capacity is 10 joules, then the energy requirement of the connection generated from this node at time t will be $\frac{3}{4} \times 10 = 7.5$ joules.

There are two criteria which will be measured: The first one is the network lifetime. An ad hoc network will be dead whenever any node cannot send data to its destination. This case happens when any node runs out of energy, making the network partitioned. The second one is the energy consumption when the network is flooded with routing control messages. The simulation results are shown in Figures 6-9, in which we use the labels as follows: AODV-MH, AODV-MMBCR, and AODV-FF&RS stand for the MH, MMBCR and our flood filtering and route selection, respectively. As the name says, all the algorithms are combined with AODV with respect to the routing discovery phase.

3.1 Network Lifetime

First, we generate 10 connections (source-destination) with different energy requirement levels. The initiating time for each connection is chosen randomly. Figure 6 shows that our proposed routing algorithm achieved the longest network lifetime in the case of 10 connections. The network lifetime of AODV-FF&RS is up to 880s, while AODV-MH and AODV-MMBCR last 660s and 810s, respectively. In our algorithm (AODV-FF&RS), the time until the first node runs

out of its energy is 630s, while those numbers in AODV-MMBCR and AODV-MH are 300s and 90s, respectively. Figure 6 also shows that the AODV-FF&RS curve is more slopping than AODV-MH and AODV-MMBCR. This means that the energy consumption is distributed more equally in case of AODV-FF&RS, making the deviation among nodes in AODV-FF&RS smaller.

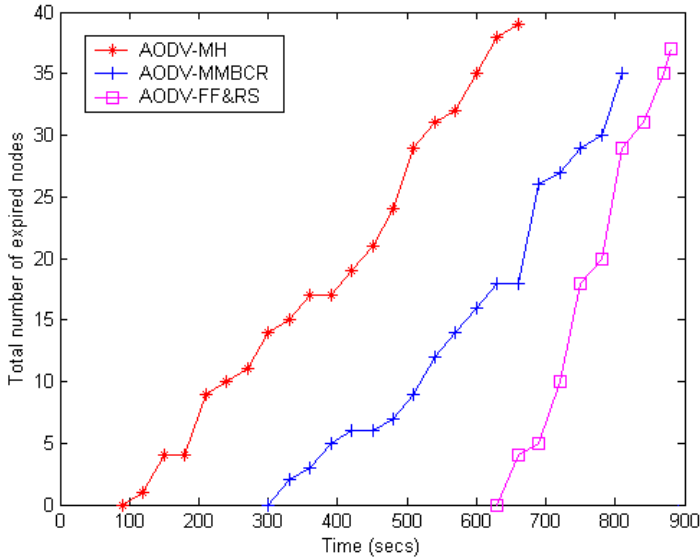


Fig. 6. The number of expired nodes as a function of simulation time (10 connections).

For the numbers of connections, 15 and 20, Figures 7 and 8 show that the network lifetime becomes shorter. In the case of 15 connections, the network lifetime of APDV-FF&RS is 660s while those of AODV-MMBCR and AODV-MH are 600s and 450s, respectively. This means that the network lifetime of our algorithm is longer than AODV-MMBCR and AODV-MH by 10% and 46%, respectively. With 20 connections, these increments are 11% and 56%. This convinces us that the more traffic load (number of connections) the better effect that our algorithm gets.

As mentioned above, an ad hoc network is dead whenever any node cannot send data to its destination. In the cases of 10 and 15 connections, the simulation results show that, although the network is dead, there are still some alive nodes with AODV-MMBCR and AODV-FF&RS. This means that when the network is partitioned, the still-living nodes cannot send and receive data. However, in the case of 20 connections, the ad hoc network is dead, because of all nodes running out of energy.

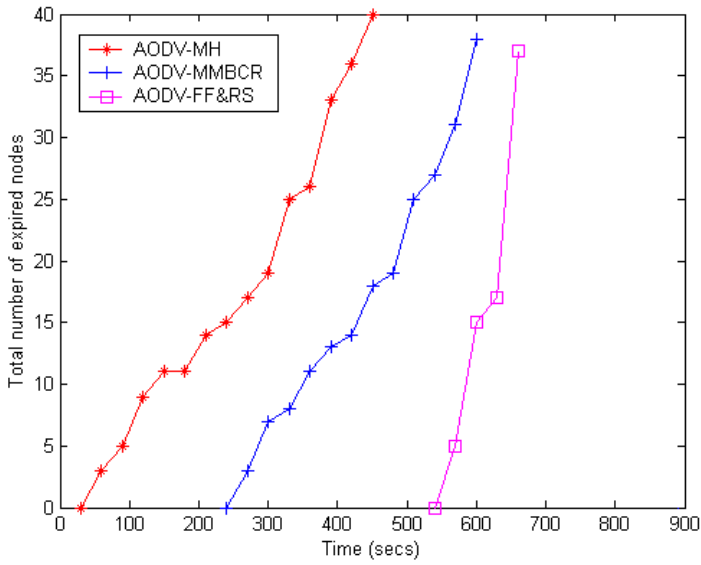


Fig. 7. The number of expired nodes as a function of simulation time (15 connections).

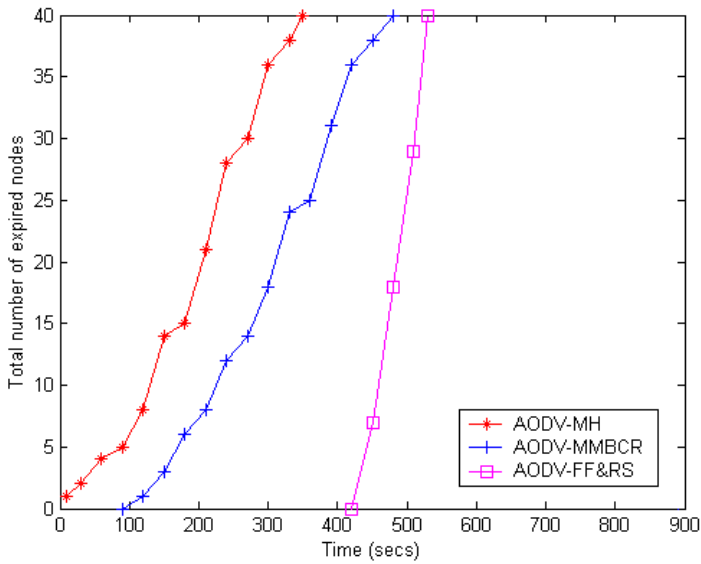


Fig. 8. The number of expired nodes as a function of simulation time (20 connections).

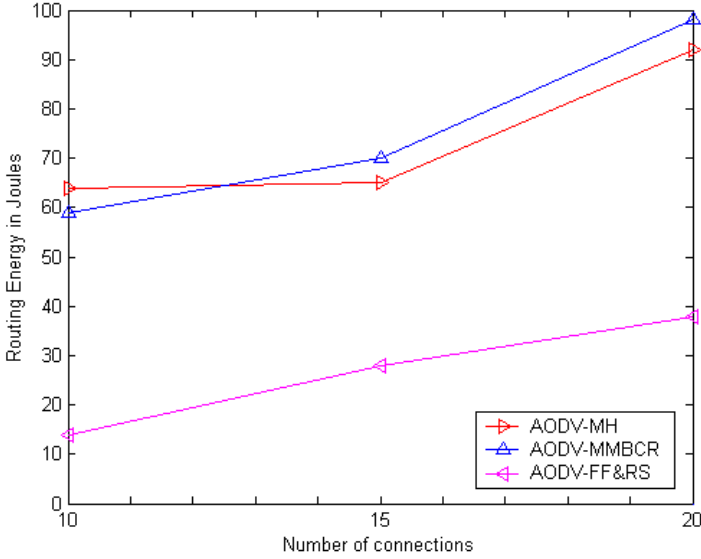


Fig. 9. Routing energy consumption.

3.2 Routing Energy Consumption

To evaluate the effectiveness of new flooding algorithm, we measure the total energy consumption for flooding routing control messages in the cases of 10, 15 and 20 connections. In each case, we calculate the total energy that all nodes have to spend for sending and receiving routing control messages. Figure 9 shows that with 10 connections the energy consumption of AODV-FF&RS, AODV-MMBCR and AODV-MH are 14, 59, 64 joules, respectively. With 15 connections, they are 28, 70, 65; with 20 connections, 38, 98, 92 joules. It means that our algorithm can save energy consumption in the routing discovery phase, from 60% upto 78%, compared to other mentioned algorithms.

4 Concluding Remarks

The lifetime of an ad hoc network can be increased by efficiently and wisely controlling the power consumption of each individual node. In this paper, we propose an adaptive routing algorithm that can help control the flooding of routing control messages and give all nodes active right to save their energy. In our approach, every node can decide by itself whether to take part in a connection section or not by performing a set of prediction rules. These prediction rules are based on energy requirement by each connection, energy availability at each node, and neighbor nodes' status. Finally, the route selections algorithm enable the destination node to choose the most suitable routing path, which has enough

energy capacity and small energy consumption. Our simulation results show that the solution can significantly last the network lifetime. Moreover, we see that it is also easy to implement by modifying an existing routing protocol such as AODV. This encourages us to implement our algorithm on a real testbed, which is our current research focus.

Acknowledgements. This research is supported by the ITRC program, Ministry of Information and Communications of Korea.

References

1. R. Jäntti and S.-L. Kim: Energy-efficient routing in wireless ad hoc networks under mean rate constraints, in Proc. IEEE VTC-Spring, Jeju, Korea, 2003.
2. W. Cho and S.-L. Kim: A Fully distributed routing algorithm for maximizing lifetime of a wireless ad hoc network, in Proc. IEEE WCNC, Stockholm, Sweden, 2002.
3. S. Singh, M. Woo, and C. S. Raghavendra: Power-aware routing in mobile in mobile ad hoc networks, in Proc. ACM MobiCom, Dallas, Texas, USA, 1998.
4. C. K. Toh, H. Cobb, and D. A. Scott: Performance evaluation of battery-life aware routing schemes for wireless ad hoc networks, in Proc. IEEE ICC, Amsterdam, Netherlands, 2001.
5. V. Rodoplu, T. H. Meng: Minimum energy mobile wireless ad hoc networks, Selected Areas in Communications, IEEE Journal on, Volume: 17 Issue: 8, Pages: 1333–1344, 1999.
6. M. Maleki, K. Dantu and M. Pedram: Power-aware source routing protocol for mobile ad hoc networks, in Proc. ISLPID, Monterey, California, USA, 2002.
7. Y. X. John, H. D. Estrin: Geography-informed energy conservation for ad hoc routing, in Proc. ACM MobiCom, Rome, Italy, 2001.
8. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris: An Energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, in Proc. ACM MobiCom, Rome, Italy, 2001.
9. C. E. Perkins, M. Belding-Royer, and D. Samir: Ad hoc on demand distance vector (AODV) routing, IETF Internet draft, draft-ietf-manet-aodv-10.txt, 2002.
10. D. Johnson and D. Maltz: Dynamic source routing in ad hoc wireless networks, Mobile Computing, edited by T. Imielinski and H. Korth, Kluwer Academic Publishers: Chapter 5, pages 153–181, 1996.
11. S. Y. Ni: The broadcast storm problem in a mobile ad hoc network, in Proc. ACM MobiCom, Seattle, Washington, USA, 1999.
12. Y. Sasson, D. Cavin, A. Schiper: Probabilistic broadcast for flooding in wireless mobile ad hoc networks wireless communications and networking, in Proc. IEEE WCNC. New Orleans, Louisiana, USA, 2003.
13. Y. Yunjung, M. Gerla: Efficient flooding in ad hoc networks, a comparative performance study, in Proc. IEEE ICC, Anchorage, Alaska, USA, 2003.
14. S. Min-Te, L. Ten-Hwang: Location aided broadcast in wireless ad hoc network systems, in Proc. IEEE WCNC, Orlando, Florida, USA, 2002.
15. C.R. Lin and M. Gerla: Adaptive Clustering for Mobile Wireless Networks, Selected Areas in Communications, IEEE Journal, Volume: 15 Issue: 7, Pages: 1265–1275, 1997.
16. The Network Simulator – <http://www.isi.edu/nsnam/ns/>, DARPA, 1995.

High Throughput Route Selection in Multi-rate Ad Hoc Wireless Networks

Baruch Awerbuch, David Holmer, and Herbert Rubens

Johns Hopkins University, Baltimore MD, USA
{baruch, dholmer, herb}@cs.jhu.edu

Abstract. Modern wireless devices, such as those that implement the 802.11b standard, utilize multiple transmission rates in order to accommodate a wide range of channel conditions. Traditional ad hoc routing protocols typically use minimum hop paths. These paths tend to contain long range links that have low effective throughput and reduced reliability in multi-rate networks. In this work, we present the *Medium Time Metric* (MTM), which is derived from a general theoretical model of the attainable throughput in multi-rate ad hoc wireless networks. MTM avoids using the long range links favored by shortest path routing in favor of shorter, higher throughput, more reliable links. We present NS2 simulations that show that using MTM yields an average total network throughput increase of 20% to 60%, depending on network density. In addition, by combining the MTM with a medium time fair MAC protocol, average total network throughput increases of 100% to 200% are obtained over traditional route selection and packet fairness techniques.

1 Introduction

Ad hoc wireless networks are self-organizing multi-hop wireless networks where all nodes take part in the process of forwarding packets. One of the current trends in wireless communication is to enable devices to operate using many different transmission rates. Many current and proposed wireless networking standards have this multi-rate capability. These include the 802.11b [1], 802.11a [2], 802.11g draft, and HiperLAN2 [3] standards. The reason for this multi-rate capability stems directly from some of the fundamental properties of wireless communication.

Due to the physical properties of communication channels, there is a direct relationship between the rate of communication and the quality of the channel required to support that communication reliably. Since distance is one of the primary factors that determines wireless channel quality, there is an inherent trade-off between high transmission rate and effective transmission range.

This range speed trade-off is what has driven the addition of multi-rate capability to wireless devices. Consumer demands for wireless devices always include both higher speed and longer range. Unfortunately a single rate represents a single trade-off point between these two conflicting goals. Since multi-rate devices support several rates, they provide a wide variety of trade-offs available for use.

This gives them a great deal of flexibility to meet the demands of consumers. This added flexibility is the primary driving force behind the adoption of multi-rate capability. It is also reasonable to assume that this type of capability will also be present in future wireless networking standards.

While multi-rate devices provide increased flexibility, they cannot change the inherent trade-off between speed and range. Both high speed and long range cannot be achieved simultaneously. Long range communication still must occur at low rates, and high-rate communication must occur at short range. This multi-rate capability merely provides a number of different trade-off points. Multi-rate devices must have protocols that select the appropriate rate for a given situation.

In infrastructure based networks, all communication takes place between nodes and access points. In this case, an additional protocol required to support multi-rate is necessary only at the medium access control (MAC) layer. Single rate nodes already have the ability to select the best access point based on the received signal strength. Thus the only additional task necessary is that of selecting the actual rate used to communicate. Since the distance between the user and the access point is dictated by the physical geometry of the network, the rate selection task must react to the existing channel conditions. In other words, the only option available to a wireless device is to select the fastest modulation scheme that works reliably.

However, this is no longer the case in ad hoc multi-hop wireless networks. In these networks, the routing protocol must select from the set of available links to form a path between the source and the destination. While in single-rate networks all links are equivalent, in multi-rate networks each available link may operate at a different rate. Thus the routing protocol is presented with a much more complex problem. Which set of trade-offs does it choose? Long distance links can cover the distance to the destination in few hops, but then the links would be forced to operate at a low speed. Short links can operate at high rates, but more hops are required to reach the destination. In addition, the path selected by the routing protocol will not only affect the packets moving along that path, but will affect the level of congestion at every node within the interference range of the path as well.

Our Contribution. We provide a general theoretical model of the attainable throughput in multi-rate ad hoc wireless networks. This model is derived from the properties of the physical and medium access control layers. The traditional technique used by most existing ad hoc routing protocols is to select minimum hop paths. These paths tend to contain long range links that have low effective throughput and reduced reliability. We present the *Medium Time Metric* (MTM) that selects higher throughput paths and tends to avoid long unreliable links. The MTM minimizes the total medium time consumed sending packets from a source to a destination. This results in an increase in total network throughput.

2 Related Work

Ad Hoc Routing Protocols. A large number of routing protocols have been proposed by the ad hoc wireless networking community. Typically these have adopted one of two major strategies: on-demand such as in AODV [4] and DSR [5], and proactive such as in DSDV [6] and OLSR [7]. The vast majority of these protocols were originally designed for single-rate networks, and thus have used a shortest path algorithm with a hop count metric (min hop) to select paths. While min hop is an excellent criteria in single-rate networks where all links are equivalent, it does not accurately capture the trade-offs present in the more complicated multi-rate networks. As ad hoc networks are likely to be deployed in multi-rate networks, it should be possible to enhance the network performance of almost any existing shortest path based protocol by adapting it to use our medium time metric.

Signal Stability Based Ad Hoc Routing Protocols. In [8] the authors show that the minimum hop path generally contains links which exhibit low reliability. In [9] and [10] the authors present routing protocols which are based on signal stability rather than just shortest path in order to provide increased path reliability. In our work, signal stability information is used not only to increase path reliability, but also to increase network throughput.

MAC Layer. Since our proposed solution is derived from properties of the MAC and physical layers, it is important to understand existing MAC layer techniques. The IEEE 802.11 standard [11] defines the most commonly used MAC protocol in ad hoc wireless networks. 802.11 based devices are used because of their widespread availability, low cost, and 802.11's ability to provide distributed medium access control when operated in "ad hoc" mode. This mode causes the stations to use the Distributed Coordination Function (DCF) protocol that operates using Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

The method of rate selection in multi-rate capable networks has been left unspecified by the 802.11 standards. As a result, several auto rate protocols have been proposed. The most commonly used protocol is Auto Rate Fallback (ARF). ARF was originally developed for Lucent's WaveLAN II devices [12], and was later enhanced for 802.11b devices [13]. ARF operates using the link level ACK frames specified by the 802.11 standard. Each node increases the rate it is using to communicate with its neighbor after a number of consecutively received acks, and decreases the rate after a number of consecutively missed acks. The advantage of this technique is that it is easy to implement because it is purely sender based, requires no modifications to the 802.11 standard.

As an alternative, the Receiver Based Auto Rate (RBAR) protocol was presented in [14]. RBAR allows the receiving node to select the rate. This is accomplished by using the SNR of the RTS packet to choose the most appropriate

rate and communicating that rate to the sender using the CTS packet. This allows much faster adaptation to the changing channel conditions than ARF, but requires some modifications to the 802.11 standard.

The Opportunistic Auto Rate (OAR) protocol, which is presented in [15], operates using the same receiver based approach, but allows high-rate multi-packet bursts to take advantage of the coherence times of good channel conditions. These bursts also dramatically reduce the overhead at high rates by amortizing the cost of the contention period and RTS CTS frames over several packets. By picking appropriate sized bursts, OAR also changes the fairness characteristic from each node sending an equal number of packets to each node getting an equal allocation of medium time. This produces a dramatic increase in overall throughput when links of multiple rates operate together in the same space. OAR also requires modifications to the 802.11 standard.

3 Network Model

Network Assumptions. This work relies on a few specific network assumptions. We assume that the ISO/OSI physical layer is capable of operating using multiple rates. We also assume that the ISO/OSI MAC layer is capable of selecting the rate used by the physical layer. In addition, we assume that the MAC layer is capable of providing information to the ISO/OSI network layer that indicates the selected rate. The network layer can then use this information to improve its routing decisions. This work stresses the importance of inter-layer communication in wireless networks.

Multi-rate Model. The multi-rate model presented in this paper is based on the 802.11b standard [1]. The topics discussed here apply to other multi-rate standards, but all examples, ranges, and rates shown in this work are based on 802.11b.

Throughout the remainder of the paper we present the results of a number of NS2 [16] simulations. In order to simulate multi-rate 802.11b, we started with the ns-2.1b7a code base and the multi-rate extensions available from the Rice Networks Group [17] that contain implementations of the RBAR and OAR protocols. The 802.11 MAC and physical wireless parameters were further modified to match the published specifications of a Lucent ORiNOCO PC Card [18], a commonly used 802.11b wireless adapter (see Table 1). Since the carrier sense (CS) threshold specification is not published, we provide an estimate. This estimate was produced by setting the difference between the carrier sense threshold estimate and the 1.0 Mbps receive threshold equal to the difference between the NS2 default carrier sense threshold (-78 dBm) and default receive threshold (-64 dBm).

Table 2 shows the ranges resulting from these simulation parameters. Real world ranges are considerably smaller due to non-zero system loss, additional noise sources, obstructions, and propagation effects beyond the simple two ray

Table 1. NS2 Simulation Parameters

Parameter	Value
Frequency	2.4 GHz
Transmit Power	15 dBm
11.0 Mbps Receive Threshold	-82 dBm
5.5 Mbps Receive Threshold	-87 dBm
2.0 Mbps Receive Threshold	-91 dBm
1.0 Mbps Receive Threshold	-94 dBm
Carrier Sense Threshold	-108 dBm
Capture Threshold	10
Propagation Model	Two Ray Ground
System Loss	0 dBm

Table 2. 802.11b Ranges

Rate (Mbps)	Maximum Range
11.0	399 m
5.5	531 m
2.0	669 m
1.0	796 m
CS	1783 m

ground model. The results presented here should be valid for any set of ranges with similar proportions regardless of magnitude.

4 Minimum Hop Route Selection

Most existing ad hoc routing protocols have utilized hop count as their route selection criteria. This approach minimizes the total number of transmissions required to send a packet on the selected path. This metric is appropriate in single-rate wireless networks because every transmission consumes the same amount of resources. However, in multi-rate networks this technique has a tendency to pick paths with both low reliability and low effective throughput.

Throughput Loss. In multi-rate wireless networks, the selection of minimum hop paths typically results in paths where the links operate at low rates. This is because the shortest path contains the fewest number of nodes between the source and destination. Fewer intermediate nodes corresponds to longer links in order to cover the same distance. Since distance is one of the primary factors that determines channel quality, the long links have low quality, and thus operate at low rates. So given the opportunity, in an effort to minimize the number of hops, shortest path selection protocols will pick paths composed of links close to their maximum range that must operate at the minimum rate.

Not only do the low link rates produce a low effective path throughput, but as a result of the shared wireless medium, this path selection degrades the performance of other flows in the network. This occurs due to the large amount of medium time required to transmit a packet at a slow link speed. All nodes within interference range of the transmission must defer while it takes place. Thus, slow transmissions reduce the overall network throughput by consuming a large amount of medium time.

Reliability Loss. Multi-rate wireless devices are inherently designed to deal with changes in connectivity due to mobility and interference. The devices provide multiple link speeds to accommodate fluctuations in link quality. In 802.11b,

as two nodes move in opposite directions, the auto rate protocol will gracefully reduce their link speeds from 11 Mbps down to 1 Mbps before they are finally disconnected.

Minimum hop path route selection has a tendency to choose routes that utilize the lowest link speed, leaving the auto rate protocol no flexibility in dealing with channel quality fluctuations. As a result, routes are often established between nodes that are on the fringe of connectivity. This occurs when nodes are able to receive broadcast transmissions, but data/ack packets are unable to be successfully delivered. While routing broadcasts are typically extremely small in size, data packets typically occupy the full frame size, making them more susceptible to corruption at high bit error rates (BER). This tendency is even further exaggerated by the way 802.11 handles broadcast transmissions as opposed to unicast transmissions. While broadcasts are sent as a single frame, unicasts require a full RTS-CTS-DATA-ACK exchange for successful delivery, which is more likely to be disrupted by a low quality channel. The end result is that small broadcasts can often be delivered even when data communication is not possible.

5 General Model and Optimality Analysis

There is some ambiguity in the literature regarding what constitutes an optimal solution for the routing problem in multi-hop wireless networks. One of the main reasons for this is the inherent difficulty in modelling the complex environment of wireless multi-hop networks. We provide a model that captures many of the effects present in such a network.

5.1 General Model of Attainable Throughput

In this work, we ignore packet scheduling issues and consider a steady-state flow model. In this model, each network edge may be fractionally shared by several flows; however, the sum of shares cannot exceed 100%. Our model of the wireless network is defined by a *transmission graph* and *interference graph*.

The *transmission graph* is defined as $G(V, E, \rho)$. V is defined as the set of nodes in the network. A transmission edge $(u, v) \in E$ if node u is capable of transmitting to node v . ρ is a function that assigns a transmission rate to each transmission edge $\rho : E \rightarrow R^+$. $\rho(e) = \hat{\rho}$ where $\hat{\rho}$ is the maximum flow rate obtainable over edge e when no other traffic exists in the network. $\hat{\rho}$ should take into account any sources of overhead such as contention, headers, and multiple frame exchanges, and represents the “real” capacity of edge e . In this general definition, the transmission graph may be directed, and the transmission rate in the reverse direction of a bi-directional edge may be different than that in the forward direction. This is possible in real wireless networks because of different node configurations and asymmetric channel effects.

The *interference graph* is defined as $G(\tilde{V}, \tilde{E})$. We define the vertices of the interference graph to be the edges of the transmission graph, so $\tilde{V} = E$. An edge

in the interference graph represents the interaction between packets transmitted on nearby transmission edges. $((a, b), (c, d)) \in \tilde{E}$ if $(a, b), (c, d) \in E$ and if a transmission on (a, b) interferes with a transmission on (c, d) .

In the general case, modelling the interference graph of an arbitrary network may be quite difficult due to complex propagation effects caused by obstacles and reflections. However, in the open space simulation configuration used in this, and many other papers, modelling the interference graph is much simpler. In this open space environment, the interference graph includes “edges” between each possible transmission edge, and all other transmission edges with an endpoint within carrier sense range of one of the transmission edge’s endpoints. This roughly corresponds to everything within a two hop neighborhood of a transmitting node.

Given the interference graph, we can define the interference neighborhood of any given edge (u, v) as follows.

$$\chi(u, v) = \{(u, v)\} \cup \{(x, y) : ((x, y), (u, v)) \in \tilde{E}\} \quad (1)$$

Consider a set of i flows, where each flow ϕ_i originates from source s_i and is sinked by receiver r_i . Without loss of generality, we can represent each flow as a sum of path flows (indexed by j).

$$\phi_i = \sum_j \phi_{ij} \quad (2)$$

Each path flow ϕ_{ij} exists only on π_{ij} , where π_{ij} is a path from s_i to r_i in the transmission graph. In other words, $\phi_{ij}(x, y)$ equals the magnitude of the path flow $|\phi_{ij}|$ if the edge lies on its path, $(x, y) \in \pi_{ij}$, or zero otherwise. Thus we have effectively decomposed the general flow ϕ_i which may traverse multiple paths simultaneously into a set of flows ϕ_{ij} that each traverse only a single path, but sum to the original flow ϕ_i .

With this setup, we can now specify a flow constraint that captures the phenomena discussed above. For each edge (u, v) in the transmission graph, the sum of the fractional shares used by all flows in the interference neighborhood of (u, v) must be less than or equal to 100%. This is a more complicated version of the classic edge capacity flow constraint.

$$\sum_{(x, y) \in \chi(u, v)} \sum_{i, j} \left(\frac{\phi_{ij}(x, y)}{\rho(x, y)} \right) \leq 1 \quad (3)$$

In this general case, Linear Programming (LP) methods are required to achieve an optimal thruput solution. Opportunity-cost based approximations are possible in both the off-line case [19] (all connections are known ahead of time) and in the online case [20,21]. Single path solutions are even harder to achieve as they require integer LP approaches.

5.2 Optimal Routing Assuming a Complete Interference Graph

Consider the special case of the general model where the interference graph is a clique (completely connected graph), i.e. each node can carrier sense each other node. In this special case, the constraint can be simplified since the interference neighborhood of any edge $\chi(u, v)$ is the same and consists of every edge in the transmission graph. In this case we wish to show the following theorem:

Theorem 1. *In the case of a complete interference graph in the stated multi-rate ad hoc wireless network model, a routing protocol that chooses a single path that minimizes the sum of the transmission times optimally minimizes network resource consumption, and optimally maximizes total flow capacity.*

Given the complete interference condition, we can rewrite the general flow constraint.

$$\sum_{(x,y) \in E} \sum_{i,j} \left(\frac{\phi_{ij}(x,y)}{\rho(x,y)} \right) \leq 1 \quad (4)$$

We can reverse the order of summation.

$$\sum_{i,j} \sum_{(x,y) \in E} \left(\frac{\phi_{ij}(x,y)}{\rho(x,y)} \right) \leq 1 \quad (5)$$

We can also decompose $\phi_{ij}(x,y)$ by moving its magnitude out of the inner sum, and changing the inner sum to include only non-zero terms.

$$\sum_{i,j} \left(|\phi_{ij}| \cdot \sum_{(x,y) \in \pi_{ij}} \left(\frac{1}{\rho(x,y)} \right) \right) \leq 1 \quad (6)$$

Since $\rho(x,y)$ was defined as the real capacity of transmission edge (x,y) , we can define the transmission time used by a unit of flow on this edge to be the inverse of this capacity.

$$\tau(x,y) = \frac{1}{\rho(x,y)} \quad (7)$$

Thus the final constraint equation becomes

$$\sum_{i,j} \left(|\phi_{ij}| \cdot \sum_{(x,y) \in \pi_{ij}} \left(\tau(x,y) \right) \right) \leq 1 \quad (8)$$

In other words, the flow over each sub path consumes a certain fraction of the capacity. The sum of these fractions must be less than one. The fraction consumed by each sub path is equal to the amount of flow on that path times the sum of the transmission times along that path. The magnitude of flow on a sub path, $|\phi_{ij}|$, will be maximized when the sum of the transmission times along that path, $\sum_{(x,y) \in \pi_{ij}} \tau(x,y)$, is minimized. Therefore, a routing protocol that selects

paths that minimize the sum of the transmission times maximizes the flow along those paths. Also, it is only necessary for each flow to have a single sub path that minimizes the sum of the transmission times, because any other sub paths will be at best equivalent to the minimum, and thus offer no additional flow capacity. Even if a flow does not use its maximum available capacity, minimizing the path transmission time minimizes the flow's consumption of the common network resource and allows other flows to increase. Thus we have shown Theorem 1 to be true.

6 Medium Time Metric

We propose a *medium time metric* (MTM) that is designed to allow any shortest path routing protocol to find throughput optimal routes assuming full interference. The MTM assigns a weight to each link in the network that is proportional to the amount of medium time used by sending a packet on that link. The weight of any given path is thus a sum that is proportional to the total medium time consumed when a packet traverses the whole path. As a result, shortest path protocols that use the medium time metric find paths that minimize the total transmission time.

We have shown that the MTM is globally optimum in the case of complete interference. In real networks the interference graph is primarily determined by the carrier sense range. While the carrier sense range is not infinite, in 802.11b networks it is greater than twice the maximum transmission range. Therefore, full interference networks are limited to four or less maximum length hops. While these small networks are useful for some applications, many applications require larger networks.

Once we consider larger networks, we can no longer claim that the MTM is globally optimal because traffic patterns and congestion may shift the optimal routes, and in very large networks multiple "disjoint" paths may be used. However, the MTM still exhibits desirable characteristics in these larger networks. If we consider a reduced class of single flow (non-congestion sensitive) single path algorithms, we find that the MTM continues to perform optimally up to a much longer path length limit. This occurs because the sum of the medium times is still an accurate predictor of total path throughput, even for paths much longer than the complete interference case. The MTM will continue to be accurate as long as one link in the path is in interference range of all other links in the path. This occurs with paths of up to seven maximum length hops, and also corresponds to the length where pipelining begins to occur (see Figure 1). Once the paths are long enough to exhibit significant pipelining, the MTM begins to underestimate their throughput potential.

We have shown that the MTM performs completely optimally in small full interference networks, and that the MTM selects optimal non-congestion sensitive single path routes for lengths up to the pipelining distance. The reader should note that this second property places no restriction on the total size of the network, which may be extremely large, and only restricts the length of the

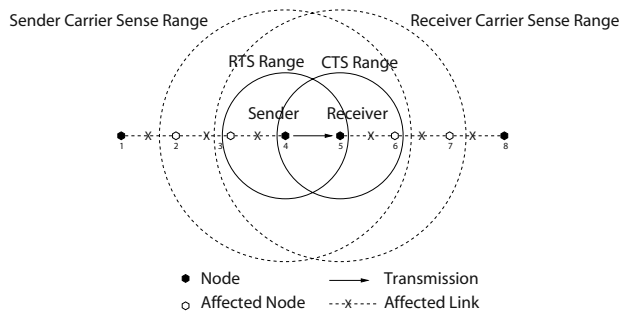


Fig. 1. Link Interference Range

actual communication paths. This is an important observation because prior research has shown that ad hoc networks scale to large sizes only if the traffic patterns remain local [22]. A local traffic pattern, such as when every node accesses the nearest Internet gateway, provides a natural path length limit allowing the MTM to operate in its optimal regions even in very large networks. The reason that non-local traffic patterns do not scale is that even with a globally optimal routing protocol, the attainable throughput at pipelining distance and beyond is extremely small, thus communicating over these distances consumes a large quantity of medium time with little gain.

6.1 Computing Link Weights

Our medium time metric states that paths that minimize the total consumed medium time should be selected. In order to accomplish this using existing shortest path protocols, we must assign a weight to each link that is directly proportional the medium time consumed by sending a packet across that link. The initial obvious solution is to use weights that are inversely proportional to the rate of the link. Using this scheme, if an 11 Mbps link was assigned a weight of 1, then a 1 Mbps link would be assigned a weight of 11 (see Figure 2).

However, we find that inverse rate weights do not accurately predict the amount of medium time consumed when sending a packet because they because they do not accurately represent an 802.11b packet transmission exchange. In 802.11b a packet is typically transmitted using an idle contention period and a four frame MAC level exchange (RTS, CTS, DATA, ACK). Much of this exchange takes place at the 1 Mbps base rate, so a large nearly constant amount of medium time is consumed by per packet MAC overhead regardless of the actual link rate. This overhead becomes a large fraction of the total consumed medium time at the higher rates, because the actual data payload transmission time becomes small (see Figure 2). This overhead is why two nodes never achieve anywhere close to 11 Mbps of real throughput over an 11 Mbps link. For example, inverse weights would select a path of ten 11 Mbps links over a single 1 Mbps link. However, a 1 Mbps link is faster (and therefore consumes less medium time) than ten 11 Mbps links.

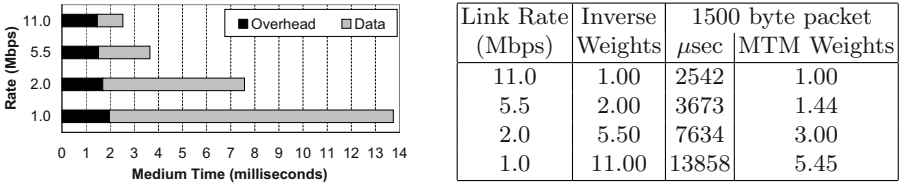


Fig. 2. Medium Times and Weights for 802.11b Transmissions

The almost fixed amount of medium time overhead caused by 802.11b introduces a dependency on packet size into our protocol. For example, the transmission time of a small packet will be dominated by the MAC overhead and will be almost the same regardless of the link rate. The implication of this phenomena is that the medium time metric would ideally use different link weights for each different packet size. This should be fairly easy to implement in link state protocols because they already have the topology information necessary to compute alternate routes using different sets of weights. However, this would be much more difficult for distance vector protocols, which require additional communication overhead for each additional set of weights. While it may be worth while in some networks to track more than one set of weights, usually the bulk of data transferred in a given network is of a single size.

An implementation of the MTM for a distance vector protocol should be tuned for the dominant packet size used by the network. This is accomplished by using link weights that are proportional to the medium time used by packets of the tuned size. These tuned weights represent the best trade-off point between short low-rate paths and long high-rate paths for packets of the tuned size. Packets that are much larger than the tuned size may have been better off traversing a longer path with even higher rate links. Similarly, packets much smaller than the tuned size may be better off taking paths that are shorter but with lower rate links. Performance of the MTM should not be significantly affected by transmissions of packets larger and smaller than the tuned size as long as those packets do not consume a large fraction of the total medium time.

In this work, the tuned packet size was chosen to correspond to a 1500 byte IP packet. This size is representative of the majority of the data transferred by the Internet [23] and corresponds to the standard Ethernet maximum transferable unit (MTU) [24]. This size was chosen over the larger native MTU of 802.11b (2314 bytes) because wireless networks today are mostly used to provide mobile access to LAN and Internet resources. In this environment, packets that flow over fixed links as well as wireless links would be limited to a 1500 byte path MTU. Purely peer to peer wireless networks would be free to use the native MTU, and could gain an additional measure of throughput due to the increased ratio of data to overhead in each packet.

Figure 2 shows the expected medium times, and corresponding proportional weights, for each rate computed according to the 802.11b standard specifications. These weights are significantly different then the inverse weights. The times are

calculated assuming a full RTS, CTS, DATA, ACK exchange. All information is sent at the base 1 Mbps rate except for the contents of the data and acknowledgement frames, which are sent at the chosen link rate. These computed times also include an estimate of the time spent backing off during contention. We used the value of half the minimum contention window size multiplied by the slot time (310 μ sec). This estimate was derived from the average time spent in the single sender case, but should function sufficiently for multiple senders. When we have an increased number of senders contending, the average idle medium time should decrease dramatically because the time spent for any particular packet is the minimum of all the senders random back offs. However, the probability of a collision also increases so the average time wasted while performing contention should not change as much as we might expect.

Even though a large number of acknowledgement packets are present in the network when TCP is used, the time total consumed by these packets is small in comparison to the data. This is particularly true when the delayed acknowledgement option of TCP is used which effectively halves the number of acknowledgements. OAR further reduces the proportion of time consumed by acknowledgement packets sent at high-rate by amortizing much of the contention and control overhead over several packets.

Since the OAR protocol significantly changes the MAC layer packet exchange, the expected medium time consumed by a packet at a given rate changes significantly. Thus in networks where OAR, or a significantly different MAC exchange, is used, different MTM weights must be calculated to match the change in consumed medium time.

6.2 Advantages

The medium time metric has several advantages over other possible routing strategies. One of its primary advantages is its simplicity. As a shortest path metric, it can be incorporated into existing distance vector or link-state protocols. The majority of existing wireless ad hoc routing protocols fall into these categories (AODV, DSR, OLSR, DSDV). It would be much more difficult to incorporate the MTM into protocols that use routing strategies other than shortest path, such as TORA [25].

The medium time metric also sidesteps the most serious problems exhibited by the optimal solution under the general model. MTM protocols only need to track changes in link rates as opposed to changes in utilization. This results in drastically lower protocol overhead. Also, there is no danger of route oscillation because MTM routes do not depend on traffic patterns. Finally, there is no danger of disrupting higher level protocols such as TCP due to out of order packet delivery because the MTM selects a single path.

Another interesting property of MTM paths is that since they naturally avoid low-rate links, they exhibit some of the properties of signal stability based routing protocols. Nodes connected by a high-rate link must move a considerable distance before the link breaks. As the nodes move further apart, the auto rate protocol reduces the link speed. As a result, proactive routing protocols, which continually

update their paths based on the MTM will naturally avoid path failures by continuously switching to higher rate links.

6.3 Discussion

Increased Hop Count. Typically, the MTM selects paths that have a greater number of hops than the minimum. While these higher rate hops consume less total medium time than the minimum number of hops, the increased number of senders could cause other detrimental effects. For instance, the increased number of senders creates higher contention for the medium. If this higher degree of contention causes a significant degradation in the throughput of the underlying MAC protocol, then the efficiency of the MTM will be degraded. The authors of [26] specifically explore this contention issue. Their paper shows that when RTS/CTS is used and the packet sizes are large (1000 bytes), the throughput of the 802.11 MAC is only reduced by 6% with 100 contending nodes. Furthermore, if the authors' proposed model based frame scheduling scheme is used, even the relatively small throughput reduction in this case is virtually eliminated. Therefore we would not expect the effect of increased contention to significantly affect the MTM when RTS/CTS is used.

An additional result of increased hop count is that there are more interface queue buffers along the path a packet must traverse. This increased amount of buffering may lead to an increase in end-to-end latency when the network is congested. While trading end-to-end latency for increased throughput is completely appropriate for bulk data transfer applications, this is not the case for delay sensitive traffic. Priority queues should be used on the intermediate nodes regardless of the routing metric used. This eliminates the need to wait in line at multiple buffers. It is also important to realize that although a min hop path may seem appropriate for delay sensitive traffic, it may actually take longer to deliver a packet over the min hop path than an MTM path. This is because it takes longer for a non-zero sized packet to be delivered across a low-rate link as opposed to a high-rate link. Many types of delay sensitive traffic, such as Telnet, use relatively small packet sizes. Small delay sensitive packets would benefit from MTM routes tuned for small packet sizes, or an MTM implementation which tracks multiple packet sizes.

Effect of Density. Routing protocols that use the medium time metric choose paths that minimize the total consumed medium time. We have argued that these paths should yield significant throughput gains when compared with minimum hop paths. However, this assumes that a path exists that utilizes less medium time than the minimum hop path. This may not be the case. Whether a better MTM path exists depends solely on the current network topology. In general, the likelihood of there existing a smaller medium time path increases as the density of the network increases.

When the density of the network is low, the topology becomes sparsely connected. This yields few choices for routing protocols to select from. In this situation, MTM and min hop will tend to pick the same path. Conversely, as the

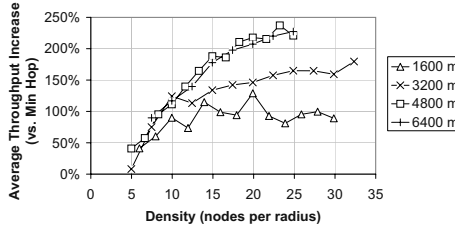


Fig. 3. Average throughput increase of MTM along a randomized straight line path.

network density increases, the abundance of nodes creates a dense, heavily interconnected topology. Routing protocols are provided with a multitude of paths from which to choose. This large number of choices allows the natural tendencies of each metric to be expressed fully.

We have constructed a simple experiment designed to illustrate the relationship between density and the performance of the MTM. A variable number of nodes are randomly placed along a straight line path of fixed length. A single UDP flow is setup between the source and destination, which are placed at opposite ends of the line. Figure 3 shows the relative throughput of the MTM and min hop routing protocols as the number of nodes and the line length are varied. The vertical axis shows the percent increase in achieved throughput over the min hop path when using the MTM. The horizontal axis shows the normalized density of the topology. We define the *normalized density* as the average number of nodes within the maximum transmission range of a given node.

The results show a clear relationship between node density and increased throughput. As expected, at low densities we see low increases as both the MTM and min hop metric pick nearly the same path. As the density increases, we see the full potential of the MTM revealed. The MTM path yields greater than three times (+200%) the throughput of the min hop path with the higher densities and longer path lengths. Longer paths yield more increased throughput than shorter paths because the MTM path utilizes the extra medium time available in long paths (from spatial reuse) much more efficiently than the min hop path.

7 Simulation Results

The purpose of this section is to evaluate the techniques proposed in this paper in a full simulated network environment. We explored the throughput gains provided by both our proposed medium time metric (MTM) and the temporally fair opportunistic auto rate (OAR) protocol over the traditional minimum hop (min hop) metric and the packet fair receiver based auto rate protocol (RBAR).

In order to implement the MTM we modified the DSDV routing protocol [6]. DSDV was selected because it is a simple example of a distance vector based proactive protocol. A distance vector based protocol is desirable in order to show that tuning for a single packet size works in a full network. A proactive

protocol was chosen in order to allow the MTM to fully extract the maximum throughput potential in a continually changing network environment. Typical on-demand protocols do not reroute until the path breaks, but a typical MTM path is constructed of short links and thus will allow considerable mobility before breaking. While this is in general a good feature, as the links lengthen due to mobility they will drop to lower rates and the throughput of the path will degrade. A proactive protocol will do a better job of preventing this degradation from occurring because it will not wait for the path to break, and thus will serve as a better platform to illustrate the potential of the MTM.

To enable the MTM two modifications to DSDV were required. The main modification was to change the computation of the routing metric, but an additional modification related to settling time was also required. In order to accomplish the metric change, we used signal information passed up from the MAC to predict the operational rate of links. Once the link rates are known, then integer weights 5, 7, 14, and 25 are used for link rates 11, 5.5, 2, and 1 respectively. These weights are directly proportional to the MTM weights discussed above, and allow all paths of up to 10 hops to be encoded in the single byte metric field used by DSDV.

Fixed Parameters. The wireless physical parameters given in Section 3 are used. In every simulation, a random way-point mobility model is used. Our simulations are setup for high mobility: the maximum speed is set to 20 meters per second and the pause time is set to zero seconds. In order to emulate a network under high load, we setup 20 flows of TCP traffic. We use the delayed acknowledgement option of TCP in order to reduce the medium time consumed by TCP acknowledgements. Each average gain result is computed from the gains in at least 25 random scenarios. Each scenario is created using a random number seed that generates the initial node placement and mobility pattern. The gains are computed by simulating each of the four protocol combinations (RBAR & min hop, RBAR & MTM, OAR & min hop, and OAR & MTM) in the exact same scenario, and then dividing the resulting total throughput by the base combination (RBAR & min hop). The base combination is representative of both the standard 802.11 MAC fairness model and the metric used by the majority of existing ad hoc routing protocols. This technique of computing gains prevents scenarios with high throughput from skewing the final average. Min hop results are obtained by using the standard DSDV protocol. MTM results are obtained using the modified DSDV. MTM link weights are tuned to match both the TCP traffic, which carries a 1460 byte payload in these simulations, and the selected auto rate protocol (RBAR or OAR).

Varying Parameters. The primary variable examined in this section is node density. The effect of node density on throughput gains was shown at the end of Section 6.3, but only in a simpler one dimensional line case. The central question this section hopes to answer, is how many nodes are required to reach the point where the MTM metric can increase throughput by selecting better

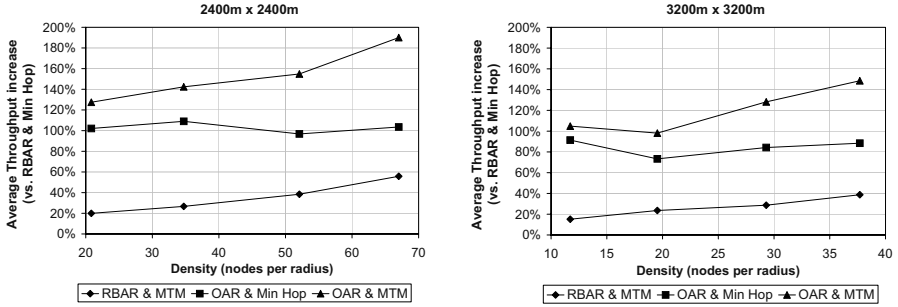


Fig. 4. Random Motion Average Throughput Gains

paths. It is clear that almost any reasonable routing metric will achieve similar performance when the network density is low because the number of available paths to choose from is limited. Since we have defined the *normalized density* as the average number of nodes within the maximum transmission range of a given node, the density in these scenarios is a function of both the number of nodes in the simulation and the total area of the simulation topology. We present simulation results for 60, 100, 150 and 193 nodes in 2400 meter by 2400 meter and 3200 meter by 3200 meter sized topologies.

Results. Figure 4 shows the average throughput gains with respect to the throughput of the RBAR & min hop combination. The average gains of the RBAR & MTM combination represent the throughput increase achieved by our proposed medium time metric under the standard packet fairness model of 802.11. As expected, we see a clear increasing trend in average gain as the density increases. Even at the lowest node density, MTM provides a modest 18% average increase. At the highest simulated density, we see a more substantial 56% average increase. The gains should continue to climb with even higher densities until a plateau is reached. Due to the increased degree of freedom in comparison to the line case, the plateau should not occur until high densities are reached.

For reference, the two sizes of simulations used in [27], a performance comparison of AODV and DSR, were 1500 meters by 300 meters with 50 nodes and 2200 meters by 600 meters with 100 nodes. Given the 250 meter nominal range used in this comparison, these simulations have the normalized densities of 21.8 and 14.9 respectively.

The average gains of the OAR & min hop combination show the throughput increase produced by the OAR protocol without changing the routing metric. As shown in the results, OAR provides quite a substantial boost in total network throughput. The gains provided by OAR come from two sources: increased overall network efficiency due to the increased proportion of time spent sending at high rates, and reduced MAC overhead due to amortization over a multiple packet burst. As a result, the OAR gains are relatively constant with respect to the node density. This experiment illustrates that the OAR protocol should

be used in high throughput multi-rate networks even if min hop is used as the routing metric.

Our analysis of the wide variety of phenomena that affect the throughput in multi-rate ad hoc wireless networks suggests that our proposed medium time metric and the OAR protocol should function well together. The MTM generally selects paths with higher rate links than the min hop, and thus gains an increased benefit from the reduced MAC overhead of high-rate links provided by OAR. Since MTM picks a greater number of high-rate links, it receives less benefit from the temporal fairness property of OAR, but is still helped in the case where paths with fast links are not available. The simulation results show that OAR and MTM do indeed function well together. The contribution of the MTM introduces the same kind of dependance on density that we saw in the pure MTM results. In the most dense simulated case, the total network throughput is almost tripled on average. These massive throughput gains lend support to the validity of both the analysis and solution presented in this paper.

8 Conclusion

In this work we have shown that minimum hop protocols tend to select paths with long slow links. As a result, these paths have low effective throughput and increase total network congestion. In addition, these paths are likely to contain long links that result in low reliability.

We have presented an improved technique for route selection in multi-rate ad hoc wireless networks. The medium time metric is proportional to the time it takes to transmit a packet on a given link. This metric selects paths that have the highest effective capacity. We have also shown the optimality of this technique under the full interference condition by presenting a formal theoretical model of the attainable throughput of multi-rate ad hoc wireless networks.

Our simulation results show an average throughput gain of 20% to 60%, depending on network density, over traditional minimum hop route selection in 802.11b networks. By combining the MTM with the Opportunistic Auto Rate (OAR) protocol, an increase of 100% to 200% is obtained over the traditional route and rate selection techniques. Our results demonstrate the importance of inter-layer communication in ad hoc routing protocol design.

References

1. *IEEE Std 802.11b-1999*, <http://standards.ieee.org/>.
2. *IEEE Std 802.11a-1999*, <http://standards.ieee.org/>.
3. *Draft ESTI EN 301 893 version 1.1.1: Broadband Radio Access Networks; HIPER-LAN Type 2*, <http://www.etsi.org/>.
4. Charles E. Perkins and Elizabeth M. Royer, *Ad hoc Networking*, chapter Ad hoc On-Demand Distance Vector Routing, Addison-Wesley, 2000.
5. David B. Johnson, David A. Maltz, and Josh Broch, *DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks*. in *Ad Hoc Networking*, chapter 5, pp. 139–172, Addison-Wesley, 2001.

6. Charles E. Perkins and Pravin Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, 1994.
7. A. Laouiti P. Muhlethaler a. Qayyum et L. Viennot T. Clausen, P. Jacquet, "Optimized link state routing protocol," in *IEEE INMIC*, Pakistan, 2001.
8. Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, and Robert Morris, "Performance of multihop wireless networks: Shortest path is not enough," in *Proceedings of the First Workshop on Hot Topics in Networks (HotNets-I)*, Princeton, New Jersey, October 2002, ACM SIGCOMM.
9. R. Dube, C. Rais, K. Wang, and S. Tripathi, "Signal stability based adaptive routing (ssa) for ad hoc mobile networks," February 1997.
10. Henrik Lundgren, Erik Nordstrom, and Christian Tschudin, "Coping with communication grey zones in ieee 802.11b based ad hoc networks," in *WoWMoM 2002*, September 2002.
11. *ANSI/IEEE Std 802.11, 1999 Edition*, <http://standards.ieee.org/>.
12. A. Kamerman and L. Monteban, "WaveLAN-II: A high-performance wireless lan for the unlicensed band," in *Bell Labs Technical Journal*, Summer 1997, pp. 118–133.
13. Anand R. Prasad and Henri Moelard, "WaveLAN-II system design note 225: Enhanced data rate control," March 1999.
14. Gavin Holland, Nitin H. Vaidya, and Paramvir Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Mobile Computing and Networking*, 2001, pp. 236–251.
15. B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," September 2002.
16. "The network simulator - ns2," <http://www.isi.edu/nsnam/ns/>.
17. "Rice networks group," <http://www-ece.rice.edu/networks/>.
18. "Orinoco wireless networks," <http://www.orinocowireless.com/>.
19. T. Leighton, F. Makedon, S. Plotkin, C. Stein, E. Tardos, and S. Tragoudas, "Fast approximation algorithms for multicommodity flow problem," in *Proc. 23rd ACM Symp. on Theory of Computing*, May 1991, pp. 101–111.
20. J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, "On-line machine scheduling with applications to load balancing and virtual circuit routing," in *Proc. 25th ACM Symp. on Theory of Computing*, May 1993, pp. 623–631.
21. Baruch Awerbuch, Yossi Azar, and Serge Plotkin, "Throughput competitive on-line routing," in *Proc. 34th IEEE Symp. on Found. of Comp. Science*. Nov. 1993, pp. 32–40, IEEE.
22. Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris, "Capacity of ad hoc wireless networks," in *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001, pp. 61–69.
23. Cooperative Association for Internet Data Analysis (CAIDA), "Analysis of NASA Ames Internet Exchange Packet Length Distributions," <http://www.caida.org/>.
24. *IEEE Std 802.3-2002*, <http://standards.ieee.org/>.
25. Vincent D. Park and M. Scott Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *INFOCOM (3)*, 1997, pp. 1405–1413.
26. Hwangnam Kim and Jennifer C. Hou, "Improving protocol capacity with model-based frame scheduling in ieee 802.11-operated wlans," in *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking*, 2003.
27. C. Perkins, E. Royer, S. Das, and M. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *IEEE INFOCOM*, 2000.

A Three-Tier Framework Supporting Soft QoS Routing in MANET*

Xin Jin, Hongbo Wang, Yaoxue Zhang, and Bin Meng

Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, China
jx01@mails.tsinghua.edu.cn

Abstract. Mobile Ad hoc Networks (MANET) is a collection of randomly moving wireless devices within a particular area. Unlike in cellular networks, there are no fixed base-stations to support routing and mobility management. Further more, many resources such as power energy and bandwidth are very limited in MANET. Concentrating on resolving these problems, we present a three-tier framework. The framework contains three cooperative algorithms, SSCA, DSRU and SQAR. SSCA is mainly responsible for topology management, DSRU is responsible for updating the routing information and the responsibility of SQAR is to select path which satisfies the QoS requirement. Experiment on the GloMoSim simulator shows that the framework proposed in this paper results in a notable reduction on energy consumption, routing overhead, packet collision times and rerouting times, and a notable improvement on network throughput and link stability, especially for the networks composed of high-speed mobile hosts.

1 Introduction

MANET is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. Numerous challenges [1] must be overcome to realize the practical benefits of MANET, because the network is highly dynamic and transmissions are susceptible to fading, interference, or collision from hidden/exposed stations.

In this paper, we concentrate on solving the problem of efficient routing caused by nodes moving with a relatively high velocity. For these fast-moving nodes, their location updates become obsolete by the time they reach the correspondent nodes. So, to get the exact position information of a mobile node needs a large routing overhead. Our overall goal is to build a system that can carry through efficient routing in such a dynamic environment, at the same time economize power consumption and maximize throughput of the network. Our overall solution is a cooperative three-tier framework. As shown in Fig.1, The framework consists of three algorithms, SSCA [2], DSRU [3] and SQAR. They have respective functions. The function of SSCA is topology management. It consists of three parts: mobility prediction, power control and

* Supported by the National Natural Science Foundation of China under Grant No.69873024; the National Grand Fundamental Research 973 Program of China under Grant No.G1998030409

clustering. DSRU is based on SSCA and it is a routing update algorithm that combines the proactive policy and the reactive policy. SQAR is on the top level of the Framework, it uses the information got from DSRU to find a path that satisfies the QoS requirement of the two communication nodes. We will introduce them separately in Section 4.

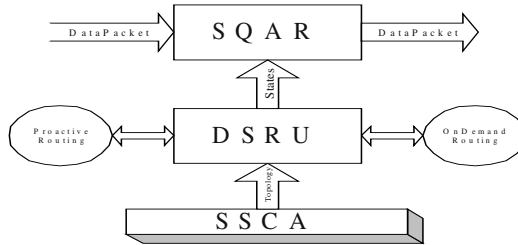


Fig. 1. System Framework

The organization of the rest of this paper is as follows. Section 2 summarizes the related works. In Section 3, we describe our framework in detail. Performance evaluation is done in Section 4. Section 5 concludes this paper and previews the future work.

2 Related Works

2.1 Mobility Prediction

As all known, mobility is the source of all difficulties. It causes frequent topological changes, makes the task of finding and maintaining routes in mobile ad hoc networks being non-trivial, and let fixed power provision be impossible. Many researchers have presented their mobility models and proposals [4, 5] to deal with this problem. These mobility models focus on the individual behavior in successive epochs, which are the smallest periods in a simulation, in which mobile hosts move in a constant direction at a constant speed. Recently, mobility prediction based on these mobility models has been reported as an effective means to decrease call-dropping probability and to shorten handover latency. J. chan et al in [6] have compared many kinds of mobility predication schemes, and concluded that the Direction Criterion has the best performance and that a high level of statistical randomness in users' movements may cause low prediction accuracy.

2.2 Power Control Scheme

Power energy is a very scarce and expensive resource for mobile hosts, and configured power transmit range of a host influences the total wireless network throughput. A recent paper [7], based on a simple interference model, derives a very interesting result. If there are N nodes in a bounded region attempting arbitrary point-

to-point communication, the throughput per node decreases at $1/N$. Obviously, it indicates that the congestion and collision control becomes more critical in larger scale cluster. The selection of optimal transmit range to maximize throughput is studied in [8, 9]. However, they do not describe any techniques for actually controlling the power, nor do they concern themselves with connectivity. Other proposals [10] aim at balancing the power consumption to prolong the life span of network, but they don't consider how to save power energy.

2.3 Clustering Scheme

Though mobility prediction and power control are important in ad hoc networks as discussed above, single scheme is not sufficient to efficient routing in ad hoc network. It is necessary to take these factors into consideration to get an integrated solution. Clustering scheme, which is easy to implement adjustment to control routing overhead and to provide stable topology, plays a crucial role in ad hoc wireless networks for efficient routing. Many clustering schemes proposed in [11, 12] are 1-hop clustering algorithms, in which every node can be reached with at most 2 hops from any other nodes in the same cluster, but there is no clusterhead. Proposal proposed in [13] tends to reelect existing clusterheads as cluster governors even when the network configuration changes.

3 A Three-Tier Framework

This section presents our framework in detail. As shown in Fig.1, the framework consists of two algorithms, SSCA, DSRU and SQAR. SSCA is a GPS based clustering mechanism. Its function is topology management, and it implements its function through three steps: mobility prediction, power control and clustering. By predicting the next location of mobile node with its historic trajectory, it adjusts the node's transmit power in advance, and controls all nodes in suitable size clusters. Based on the clusters, DSRU is proposed in order to control the routing overhead while gets relatively exact global topology information. DSRU is a hybrid routing algorithm. Its essential idea is to find a balance between optimal path and routing overhead. At last, we propose the SQAR. It is responsible for selecting and maintaining the paths that can satisfy the QoS requirement of the nodes. In the following sections, we will introduce them separately.

3.1 Suitable Size Clustering Algorithm (SSCA)

3.1.1 Mobility Prediction

By interacting with Global Position System (GPS), any host can get its location (x,y,z). In a very short period, because of the inertia effect, we can assume that the force acting on the host moving with high speed is constant, and this force can be decomposed in three dimensions, so we can also assume that the velocity variance is constant in three directions separately.

As all know the principle motion law:

$$s = v * t + \frac{1}{2} a * t^2 = \bar{v} * t \quad (1)$$

and

$$V = v + a * t \quad (2)$$

Here, S is the displacement in the period t, v is the initial velocity and α is the acceleration with same direction of v. we employ V denoting the final velocity after period t.

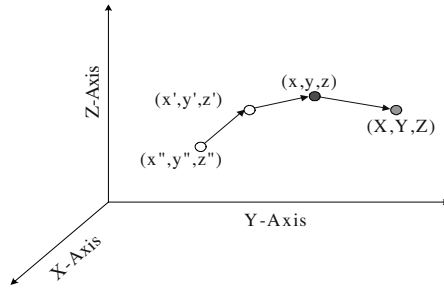


Fig. 2. Illustrating node motion trajectory, where (x'', y'', z'') and (x', y', z') are the history location. Current node locates $L(x, y, z)$. we predict it will be (X, Y, Z) in the next hits.

Fig.2 shows the trajectory of a mobile node. Now we employ v'_x, v'_y, v'_z to denote the node average motion velocity in the segment from (x'', y'', z'') to (x', y', z') and the segment from (x', y', z') to (x, y, z) in the X-axis (same mean as v_x, v_y, v_z), and T to denote location sampling cycle, then we can get (3) from (1),

$$\begin{aligned} v'_x &= \frac{x' - x''}{T}, v_x = \frac{x - x'}{T} \\ v'_y &= \frac{y' - y''}{T}, v_y = \frac{y - y'}{T} \\ v'_z &= \frac{z' - z''}{T}, v_z = \frac{z - z'}{T} \end{aligned} \quad (3)$$

(4) From (2)

$$\begin{aligned} v_x &= a_x * T + v'_x \\ v_y &= a_y * T + v'_y \\ v_z &= a_z * T + v'_z \end{aligned} \quad (4)$$

Additional, in very short slice T, we assume that the acceleration is the same as the last slice, so the next most probable location can be predicted as,

$$\begin{aligned} X &= x + (v_x + a_x * T) * T \\ Y &= y + (v_y + a_y * T) * T \\ Z &= z + (v_z + a_z * T) * T \end{aligned} \quad (5)$$

Replace v, a, and T in (5) with (3), (4), we get a simpler expression (6).

$$\begin{aligned}
X &= 3 * x - 3 * x' + x'' \\
Y &= 3 * y - 3 * y' + y'' \\
Z &= 3 * z - 3 * z' + z''
\end{aligned} \tag{6}$$

3.1.2 Power Control Scheme

Power control is a necessity in multi-hop networks, both to save power and to maximize the network throughput. In this context, we present an efficient power control scheme. As mentioned in Section 4.1.1, each node deploys a geologic method to find its physical location. Now, we derive the formula employed in SSCA to adjust the power. It is based on a well-known generic model for propagation [14] by which the propagation loss function varies as some power of distance. The value of is usually between 2 and 5, depending on the environment, specifically, if R is the loss in dB, then

$$R(d) = R(d_{thr}), \text{ if } d < d_{thr} \tag{7}$$

$$R(d) = R(d_{thr}) + 10 * \log_{10}(d/d_{thr}) \text{ if } d \geq d_{thr} \tag{8}$$

where d is the distance, and dthr is a threshold of distance below which the propagation loss is a constant; all logarithms in the remainder of this section are based on 10.

Let sc, pc, respectively, denote the current cluster size and current clusterhead transmit power. We need an expression for new transmit power p_d , so that the cluster has the desired size s_d .

Let cd_i^j , ed_i^j , respectively, denote the current distance and expected next distance from i to its neighbor j, and P_e denotes adjustment targeted power, $[d_{min}, d_{max}]$ is the range of adjustable power transmit distance.

$$\begin{aligned}
cd_i^j &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \\
ed_i^j &= \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2}
\end{aligned} \tag{9}$$

$$cd = \max\{cd_i^j\}, j \in N_i \tag{10}$$

$$ed = \min\{d_{max}, \max\{ed_i^j\}\}, j \in N_i \tag{11}$$

As noted previously, S is the receiving sensitivity for all nodes. Then as $cd > d_{thr}$ & $ed > d_{thr}$, the following hold:

$$p_c - (\Re(d_{thr}) + 10 * \epsilon * \log(\frac{cd}{d_{thr}})) = S \tag{12}$$

$$p_d - (\Re(d_{thr}) + 10 * \epsilon * \log(\frac{ed}{d_{thr}})) = S \tag{13}$$

From (12) and (13), we get a simpler equation

$$p_d = p_c - 10 * \varepsilon * \log\left(\frac{ed}{cd}\right) \quad (14)$$

In our system, we employed $\varepsilon = 4$, but ε can also be configured depending upon the environment. Equation (14) can thus be used to calculate the new power periodically. We note that the formula applies for both power increasing and decreasing to bring the cluster size close to s_d .

3.1.3 Clustering Scheme

In this section, we present a suitable size clustering scheme. Its purpose is to override the high threshold bounds and to adjust the power if the topology change is indicated by the routing update results in undesirable connectivity. It is triggered whenever an event driven or periodic link-state update arrives and it is incremental, in which it calculates new transmit power not from scratch, but being based on the currently used values.

Initially, all nodes start with the maximum possible power. With 1-CONID [15] clustering algorithm, it results in a maximally connected network, which enables successful propagation of updates and the initialization of a network topology database at each node. After this initialization, clusterheads conduct power control to maintain proper size of cluster around the configure value s_d by adjusting its pilot signal level, as follow:

$$\begin{aligned} & s_c < s_d : \\ & \quad ed = \max\{ed + \text{rand}(\Lambda), d_{\max}\} \\ & \quad \text{recalculate } p_e \text{ with (14)} \\ & s_c > s_d : \\ & \quad ed = ed_i^{s_d}, \text{ where } ed_i^1 < ed_i^2 < \dots < ed_i^3 < \dots < ed_i^{s_c} \\ & \quad ed = \min\{ed, d_{\min}\} \\ & \quad \text{recalculate } p_e \text{ with (16)} \\ & \quad \text{adjust transmit power to } p_e \end{aligned}$$

Λ is the system configurable value that is related to the power adjustment capability.

If a cluster has too many nodes including ordinary nodes (mobile stations) and gateways, the clusterhead reduces its power signal to make the area of the cluster shrink. If a cluster is suffering from isolation or has too little connectivity, its clusterhead increases power signal. Since both parties (clusterhead and mobile station) can control transmit power, a power signal should embed its transmit power level. Otherwise, the open loop power control would be impossible because the open loop control assumes a predefined power level of pilot signals.

3.2 Dynamic Self-Adaptive Routing Update Algorithm (DSRU)

Based on the established cluster, we propose a hybrid routing algorithm. It is the combination of proactive policy and reactive policy. Intra-cluster routing uses a proactive policy, whereas the inter-cluster routing is reactive. In networks with low rates of mobility, clustering provides an infrastructure, which is more proactive. This enables more optimal routing by increasing the distribution of topology information when the rate of change is low. When mobility rates become higher, cluster size will diminish and reactive routing will dominate. The hybrid policy accompanies a better balance between routing overhead and quick routing.

The routing update includes two procedures, intra-cluster routing update and inter-cluster routing update. Intra-cluster routing update cycle is shorter than that of inter-cluster routing cycle.

The intra-cluster routing update is implemented by clusterhead. The clusterhead sends intra-cluster route status packet to its cluster members periodically, and the members will update their routing table after they receive the packet.

The inter-cluster routing update can employ any proposed proactive routing schemes. The clusterhead designates some gateway nodes as inter-cluster routing updaters. These updaters execute inter-cluster routing update procedure as follow:

1. The clusterhead initiates inter-cluster route status packet and sends it to its gateway nodes periodically.
2. Any gateway node receiving a inter-cluster update packet performs below actions:
 - a) Integrates the routing status information of this packet to its local routing table and records the updating path of the source cluster.
 - b) Refreshes timers of routing table items according to the new arrival inter-cluster update packet.
 - c) Checks the travel path of this packet with that of last update. If successive update packets initiating from the same cluster have traveled on the same cluster path, the updater forwards new update packet to its direct neighbors except for the coming cluster, otherwise no forwarding is performed.
3. The timeout route items are removed from local routing table when its timer event arrives.

Fig.3 illustrates the process of above procedure. Node A initiates an inter-cluster route update packet and sends to node B transferred by node I (similar to node C and D), if node B has received inter-cluster update packet of the cluster delegated by node A in the successive inter-cluster update periods, node B forwards this update packet to node E and F, so node E and other nodes which are in the same cluster with E know the topology and link status of the cluster delegated by A. At this moment, local routing table of node E at least includes the status of three clusters, which delegated by E, B and F respectively.

3.3 Soft QoS Assurance Routing (SQAR) Algorithm

In this section, we have proposed a Soft QoS Assure Routing Algorithm. SQAR is based on SSCA and DSRU, its function is to select the path which is satisfied the QoS request of the application, guarantee the validity of the path, and balance the load.

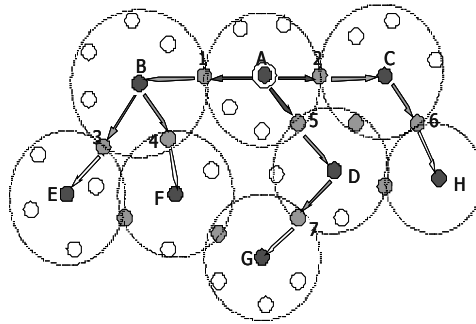


Fig. 3. Example of inter-cluster routing update

3.3.1 Route Request Procedure

When a source node wants to communicate with a destination node and has no routing information about this destination, it initiates a route-request procedure to find a route to the destination by broadcasting a route-request (RREQ) message to its neighbors as shown in Fig.4 and sets a route discovery expiration timer. The purpose of this timer is to detect whether the destination is reachable or not. The route discovery expiration time depends on the size of the network. The RREQ message has contained the following variant: *bcastId*, *destAddr*, *destSeq*, *srcAddr*, *srcSeq*, *lastAddr*, *hopCount* and *bandwidth*. The *bcastId* is incremented whenever the source node issues a new RREQ message. The intermediate node uses the pair (*srcAddr*, *bcastId*) to identify a RREQ message. When duplicate copies of RREQ arrive and their hop counts are greater than the hop count value recorded in routing table, they are discarded. The *srcSeq* number is used to maintain freshness information about the reverse route to source and *destSeq* number specifies the most recent routing information of the route to destination maintained in source node. When a node receives a RREQ message, it performs one of the following steps:

1. If the receiving node knows a route to the destination node, it checks to see if the route is current by comparing the *destSeq* in its own route entry to the *destSeq* in RREQ. If the *destSeq* in RREQ is not greater than that recorded in its own route entry and there is enough bandwidth in the path to satisfy the requirement, the intermediate node sends back a route reply (RREP) message.
2. If the receiving node does not know a route to the destination node or the *destSeq* in RREQ is greater than that recorded in its own route entry, it decreases the *hopCount* in RREQ by one. If the *hopCount* is zero, or the bandwidth of the node can't satisfy the QoS requirement in RREQ, or the role of the node is normal then the node will discard the RREQ. Otherwise, the receiving node attempts to build reverse links to the nodes that sent the RREQ message and then re-broadcast the RREQ message to their neighbors.

Each intermediate node repeats above procedure until an intermediate node finds a route to the destination, or the destination is reached. When an intermediate node knows a route to the destination, or the destination node sends RREP message back along the reverse link, the route request procedure is terminated.

For each intermediate node, after it has relayed a RREQ message, it begins to time. If it has not received a RREP message after $(2 * (Delay_{End-to-end} - Delay_{Current Request}))$, then it will re-relay the RREQ message. After several times, it will send a route-error (RERR) message to upstream node. Each intermediate node will do the same procedure until an intermediate node finds a route to node D, or node D is reached

3.3.2 Route Reply Procedure

After the route-request procedure, the RREQ message will arrive to the destination or a node that possesses a route to the desired destination. Then the receiving node will send an RREP message to the source along the reverse links. The RREP message has contain the following variant: bcastId; destAddr; destSeq; srcAddr; srcSeq; lastAddr; hopCount; bandwidth; avaBandwidth. Note that the destSeq is extracted from the RREQ message and the intermediate nodes use the pair (destSeq, destAddr) to identify a RREP message. The avaBandwidth records the max available bandwidth of the path. As the RREP message travels to the source, each node along the reverse path will perform one of the following operations.

1. If the RREP message is not a duplicate message and the receiving node is the source, then it will create a forward link, update its routing table and start communication.
2. If the RREP message is a duplicate message from another neighbor, the receiving node will set up a forward link, update its backup table, and then discard the RREP message; otherwise, it will discard the message.
3. If neither 1 nor 2 described above is true, the receiving nodes will create a forward link, update its routing table and send an RREP message back along the reverse link.

3.3.3 Route Maintenance Procedure

Once a next hop becomes unreachable, upstream nodes must perform appropriate operations to recover the routing path. In SQAR routing protocol, intermediate nodes are responsible for finding new routes when the next hops become unreachable. This can be done by maintaining multiple next-hops in each mobile host. When link failures occur during communication, upstream nodes detect the failures and eliminate invalid routes. If these upstream nodes have more than one next hop in their routing tables, they select new one, otherwise they inform their upstream nodes along the reverse links. These upstream nodes then become responsible for reconstructing new routes. Thus, the number of new route reconstructions is reduced.

4 Simulation and Performance Evaluation

4.1 Simulation Environment

We have implemented our algorithms within the GloMoSim [16] library. The GloMoSim library is a scalable simulation environment for wireless network system using the parallel discrete-event simulation language called PARSEC [17]. Our simulation models a network within 1000*1000 meter square and the nodes in the

network are placed uniformly. Radio propagation range for each node is 150 meters and channel capacity is 2 Mbits/sec. In most of experiments unless specified, the network consists of 100 nodes and the average moving speed varies from 5m/s to 45m/s. Each simulation executed for 10 minutes of simulation time. We run each scenario three times and the data collected are averaged over those runs.

To validate the effectiveness of mobility prediction, we compare the performances of our clustering algorithm in two cases. One is calculating the next transmit power range only based on the prediction distance (Abbr. as **olnp**), and the other based on the maximum of current distance and prediction distance (Abbr. as **nowp**). In order to test the advantage of power control, we have simulated the version of no power control (**noAdj**), which is 1-CONID. In addition, we also simulate **FSR** [18] algorithm for performance comparison.

4.2 Simulation Results

We first compare the packet collision, transmit power ratio among the **olnp**, **nowp**, **noAdj** and **FSR**, then we compare the routing overhead between **FSR** and **DSRU**. We will show how node density and moving speeds impact the network performance.

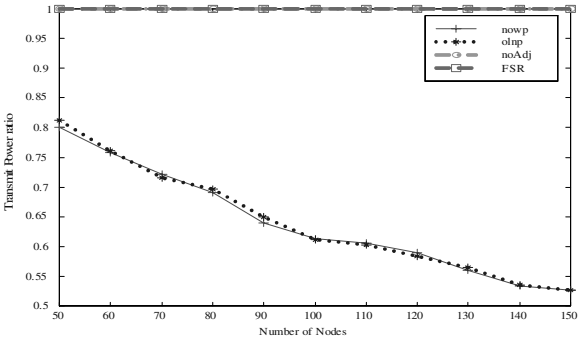


Fig. 4. (a) Transmit power ratio comparison by number of nodes

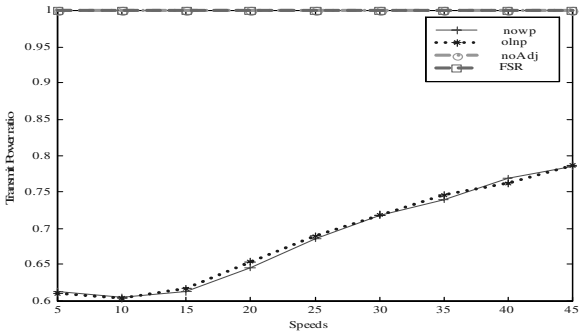


Fig. 4. (b) Transmit power ratio comparison by mobility speeds

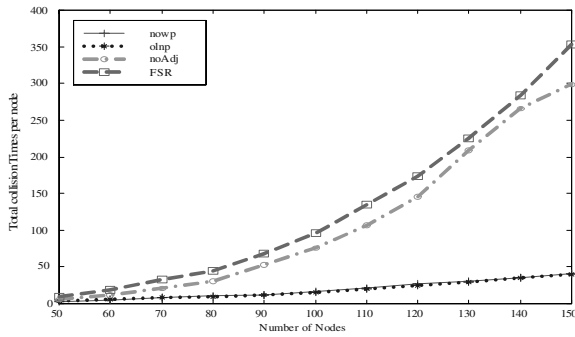


Fig. 5. (a) Packet collisions per node comparison by number of nodes

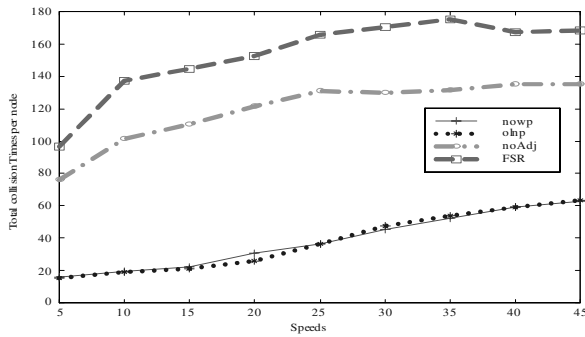


Fig. 5. (b) Packet collisions per node comparison by mobility speeds

We can see how the transmit power ratio is impacted by number of nodes and the node moving speeds from Fig.4. With the number of nodes increasing, the node density increases because the terrain is constrained in 1000mx1000m, which means average distance between two nodes is shortening. In this situation, every node shrinks its transmit power ratio. From Fig.4.(a), we can see that we will save more power energy to prolong the life span of total network. For a special ad hoc wireless network composed of 100 nodes in the specified area, we can save about 36% transmit power energy. As shown in Fig.4.(b), transmit power ratio increases as the node moving speed increasing. The reason is that higher moving speed increases the probability of current active neighbor moving away the clusterhead. For maintaining the connection, the clusterhead must increase its transmit power, then the dominated neighbor increases its own transmit power so that it keeps connected with clusterhead.

Fig.5 shows how the packet collisions increase with the number of nodes and the node moving speeds increasing. Fig.5.(a) shows that collisions of **onAdj** and that of **FSR** increase rapidly as the node density increasing due to the node propagating broadcast to the network. Fig.5.(b) shows that collisions of all the four situations increase as the node speeds increasing. Because transmit power increases as the node

moving speeds increase, more nodes will enter clusterhead covered area, so the average collisions increase.

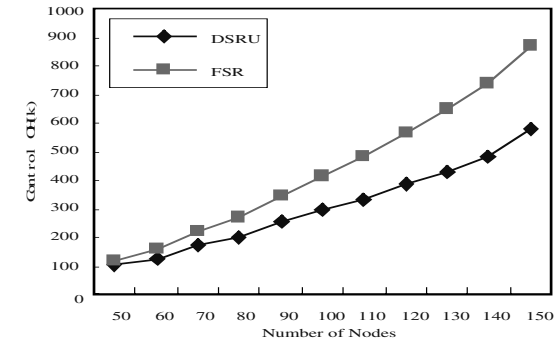


Fig. 6. (a) Control Overhead comparison by number of nodes

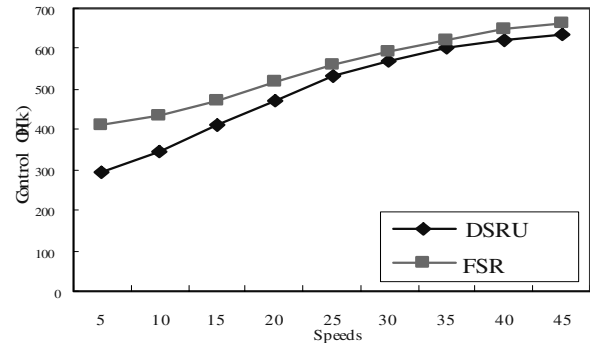


Fig. 6. (b) Control Overhead comparison by mobility speeds

Fig.6.(a) reports the comparison of routing overhead between **FSR** and **DSRU** by the number of nodes. When the number of nodes increases, the routing overhead of the both algorithms increases. But the routing overhead of **DSRU** is always lower than that of **FSR**. Because in **FSR**, every node makes itself the centre of a circle and sends routing update message to other nodes with the frequency corresponding to the scope radius, while in **DSRU**, the routing update process is based on clusters. So when the number of nodes increased only a part of new nodes participate in the routing update process. Therefore the routing overhead of **FSR** is increasing faster than that of **DSRU**. Especially when the number of node is more than 100, **DSRU** reduces more than 40% of routing overhead compared with **FSR**.

Fig.6.(b) reports the comparison of routing overhead between **FSR** and **DSRU** by the moving speeds. When the node moving speed is slow the routing overhead of **DSRU** is much lower than **FSR**. With the speed increasing, the routing overhead of **DSRU** and **FSR** are both increasing, but even in the worst case that the node moving

speed is very fast and the cluster can not be maintained, because the **DSRU** can dynamically reduce the scope of routing update, so the routing overhead of **DSRU** is still lower than that of **FSR**.

The performance evaluation of **SQAR** is under way. In the future work, we will add the results into the whole framework.

5 Conclusions

In this paper, we have proposed a cooperative three-tier framework for QoS routing in MANET. It tries to resolve the problem from three aspects: topology management, routing update and selecting path according to QoS parameter. The first two parts of the framework, **SSCA** and **DSRU**, have been implemented in **GloMoSim** simulator. The simulation results show that these two algorithms can effectively maintain the link stability between the nodes, limit the power consumption, and reduce the delay, collision times and packet loss ratio.

References

1. Chakrabarti, S., Mishra, A. QoS issues in Ad hoc wireless network. *IEEE Communications Magazine*, 2001,39(2):142–148.
2. Wang Hongbo, Zhang Yaoxue. Suitable size clustering algorithm for Ad hoc wireless networks. *Journal of Software*, September 2002.
3. Jin Xin, Wang Hongbo, Zhang Yaoxue. A Dynamic Self-adaptive Routing Update Algorithm for MANET. In the International Congerence on Communication Technology(ICCT,2003), Beijing, China.
4. Zonoozi, M., Dassanayake P. User mobility modeling and characterization of mobility patterns. *IEEE Journal on Selected Areas in Communications*, 1997,15(7):1239–1252.
5. Chiang, C.-C. Wireless network multicasting [Ph.D. Thesis]. Los Angeles: University of California, 1998.
6. Chan, J., Zhou, S., Seneviratne, A. A QoS adaptive mobility prediction scheme for wireless networks. In: Weber, J., ed. *Proceedings of the IEEE Globecom'98*. Sydney: IEEE Communication Society Publisher, 1998.
7. Gupta, P., Kumar, P.R. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 2000,46(2):388–404.
8. H.Takagi, L.Leinrock. Optimal Transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 1984, 32(3): 246–257
9. T. Huo,V.O.K.li. Transmission range control in multihop radio networks. *IEEE Transaction on Communications*, 1986, 34(1) :38–44.
10. S. Singh, M. Woo, and C. S. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. In: *Proceeding of ACM/IEEE MOBICOM '98*, New York, 1998. 181–190.
11. Taek Jin Kwon, Mario Gerla. Clustering with Power Control. In *Proceedings of IEEE MILCOM'99*, Atlantic City, NJ, 1999.
12. Chunhung Richard Lin, Mario Gerla. Adaptive Clustering for Mobile Wireless Networks. *IEEE Journal on Selected Areas in Communications*, 1997, 15(7): 1265–1275.
13. Alan D.Amis,Ravi Prakash,Thai H.F. Vuong,Dung T.Huynlh. Max_Min D-Cluster Formation in Wireless Ad Hoc Networks. In *Proceedings of INFOCOM'2000*, Tel Aviv, Israel, 2000.

14. Rappaport, T.S. *Wireless Communications, Principles and Practice*. Prentice-Hall, 1996.
15. Chen, G., Garcia, F., Solano, J., et al. Connectivity based k-hop clustering in wireless networks. In: Sprague, R.H., ed. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002)*. Hawaii: IEEE Computer Society Publisher, 2002.
16. M.Takai, L.Bajaj, R.Ahuja, R.Bagrodia and M.Gerla. GloMoSim: a scalable network simulation environment. Technical report 990027, UCLA, Computer Science Department, 1999.
17. R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y.Song. PARSEC: a parallel simulation environment for complex systems. *IEEE Computer*, vol. 31, no. 10, Oct. 1998, pp.77–85.
18. Mario Gerla, Xiaoyan Hong, Guangyu Pei.,” Fisheye state routing protocol (FSR) for ad hoc networks”, In *Internet-Draft, draft-ietf-manet-fsr-03.txt*, Jule 2002

Achieving Maximum Throughput and Service Differentiation by Enhancing the IEEE 802.11 MAC Protocol*

Bo Li and Roberto Battiti

Department of Computer Science and Telecommunications, University of Trento,
38050 POVO Trento, Italy
{li, battiti}@dit.unitn.it

Abstract. To satisfy various needs and priorities of different users and applications, Wireless LANs are currently evolving to support service differentiation. Work is in progress to define a standard enhanced version of the IEEE 802.11 Distributed Coordination Function (DCF), capable of supporting QoS for multimedia traffic at the MAC layer. This paper focuses onto one of the building blocks of this enhancement, i.e., differentiating the minimum contention window size according to the priority of different traffic categories. The novel contribution is the analysis of the optimal operation point where the maximum throughput can be achieved. The second contribution is the proposal of simple adaptive schemes which can lead the system to operate under the optimal operation point and, at the same time, achieve the target service differentiation between different traffic flows. Results obtained in the paper are relevant for both theoretical research and implementations of real systems.

1 Introduction

To provide seamless multimedia services to nomadic users and to use the spectrum in an efficient way, the “wireless mobile Internet” based on the 802.11 protocol has to provide suitable levels of Quality of Service [1]–[3]. The starting point of the paper is the IEEE 802.11 Distributed Coordination Function (DCF) standard [4], which is compatible with the current best-effort service model of the Internet, see [5]–[11] for seminal works on related models and simulations.

In order to support different QoS requirements for various types of service, a possibility is to support differentiation at the IEEE 802.11 MAC layer, as proposed in [12]–[15]. In these papers, service differentiation is achieved by assigning different minimum contention windows, different inter-frame spacing, or different maximum frame lengths to different types of traffic flows. In [16], both the Enhanced Distributed Coordination Function (EDCF) and the Hybrid Coordination Function (HCF), defined in the IEEE 802.11e draft, are extensively evaluated through simulation. In [17], the performance of the IEEE 802.11 MAC protocol with service differentiation is analyzed. However, the model is complex, which makes it difficult

* This work is supported by the project of WILMA funded by Provincia Autonoma di Trento (www.wilmaproject.org)

to obtain deeper insight into the system performance. In [18], we propose a simple analysis model to compute the throughput in a WLAN with Enhanced IEEE 802.11 DCF.

Some more practical adaptive schemes are proposed to make the system cope with the dynamic traffic. In [19], a scheme to dynamically tune the IEEE 802.11 protocol parameters has been proposed to achieve maximum throughput. However, multiple service types are not considered. In [20], an adaptive EDCF scheme is proposed. The method uses the idea of slowly decreasing the contention window size to improve the system utilization. Service differentiation is also considered but without a rigorous analysis model to achieve maximum throughput and target service differentiation at the same time. The problem of fairly sharing channel resources is considered for example in [21]-[22] for the case of non-fully connected or ad-hoc networks. Achieving efficient utilization and weighted fairness for a fully connected network is considered in [23], where a simplified uniform backoff scheme is assumed.

In the paper, we consider the more complex *standard* backoff scheme with the aim of minimizing changes of the existing and widely adopted protocol.

2 IEEE 802.11 DCF: Basic Principles and Enhancements

The basic 802.11 MAC protocol, the Distributed Coordination Function (DCF), works as listen-before-talk scheme based on Carrier Sense Multiple Access (CSMA), with a Collision Avoidance (CA) mechanism to avoid collisions that can be anticipated if terminals are aware of the duration of ongoing transmissions (“virtual carrier sense”). When the MAC receives a request to transmit a frame, a check is made of the physical and virtual carrier sense mechanisms. If the medium is not in use for an interval of DIFS, the MAC may begin transmission of the frame. If the medium is in use during the DIFS interval, the MAC selects a backoff time and increments the retry counter. The backoff time is randomly and uniformly chosen in the range $(0, W - 1)$, W being the contention window. The MAC decrements the backoff value each time the medium is detected to be idle for an interval of one slot time. The terminal starts transmitting a packet when the backoff value reaches zero. When a station transmits a packet, it must receive an ACK frame from the receiver after SIFS (plus the propagation delay) or it will consider the transmission as failed. If a failure happens, the station reschedules the packet transmission according to the given backoff rules. At the first transmission attempt, W is set equal to a value CW_{\min} called minimum contention window. After each unsuccessful transmission, W is doubled, up to a maximum value $CW_{\max} = 2^m \cdot CW_{\min}$.

The basic DCF method is not appropriate for handling multimedia traffic requiring guarantees about throughput and delay. Because of this weakness, task group E of the IEEE 802.11 working group is currently working on an enhanced version of the standard called IEEE 802.11e. The goal of the extension is to provide a distributed access mechanism capable of service differentiation [24]-[25]. In the interest of conciseness, we are interested in gaining insight into one of the building block used to achieve differentiation, i.e. differentiating the minimum contention window sizes according to the priority of each traffic category.

2.1 System Modeling

We assume that the channel conditions are ideal (i.e., no hidden terminals and capture) and that the system operates in saturation: a fixed number of traffic flows always have a packet available for transmission.

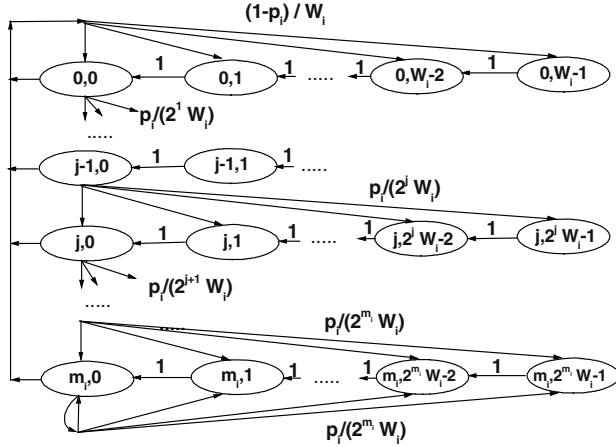


Fig. 1. Markov model of backoff process for type- i traffic

Because our analysis can be easily extended and for the sake of simplicity, only two different types of traffic are considered with n_i traffic flows for traffic of type i ($i = 1, 2$). Moreover, it is assumed that each mobile terminal has only one traffic flow. Let $b_i(t)$ be the stochastic process representing the backoff time counter for a given traffic flow with type i . Moreover, let us define for convenience $W_i = CW_{\min,i}$ as the minimum contention window for traffic type i . Let m_i , “maximum backoff stage” be the value such that $CW_{\max,i} = 2^{m_i} \cdot W_i$. Let $s_i(t)$ be the stochastic process representing the backoff stage $(0, 1, \dots, m_i)$ for a given traffic flow with type i .

We use a two-dimensional discrete-time Markov chain to model the behavior of a traffic flow with type i . The states are defined as the combinations of two integers $\{s_i(t), b_i(t)\}$. The Markov chain for type- i traffic flows are shown in Fig. 1. All details about the analysis can be found in [10], [18] and [29].

2.2 Throughput Analysis

Let $q_i(j, k)$, $j \in [0, m_i]$ and $k \in [0, 2^j \cdot W_i - 1]$, be the stationary distribution of the chain. It is easy to find that

$$q_i(0, 0) = (2(1 - 2p_i)(1 - p_i)) / ((1 - 2p_i)(W_i + 1) + p_i W_i [1 - (2p_i)^{m_i}]) \quad (1)$$

τ_i is defined as the probability that a station carrying type- i traffic transmits in a randomly chosen slot time. We have

$$\tau_i = \sum_{j=1}^{m_i} q_i(j,0) = (2(1-2p_i)) / ((1-2p_i)(W_i+1) + p_i W_i [1 - (2p_i)^{m_i}]) \quad (2)$$

With the above probabilities defined, we can express packet collision probabilities p_i as:

$$p_i = 1 - (1 - \tau_i)^{n_i-1} \prod_{j=1, j \neq i}^L (1 - \tau_j)^{n_j} \quad (3)$$

After combining equations (2) and (3) and by using Successive Over-Relaxation (SOR) numerical method [26], we can get all the values for p_i and τ_i .

Moreover, we define $Q(i, j)$ as the probability that there are a number i of type-1 stations and a number j of type-2 stations transmitting within a randomly selected slot. Then, we have

$$Q(c_1, c_2) = \binom{n_1}{c_1} \cdot \tau_1^{c_1} (1 - \tau_1)^{n_1 - c_1} \cdot \binom{n_2}{c_2} \cdot \tau_2^{c_2} (1 - \tau_2)^{n_2 - c_2} \quad (4)$$

The normalized system throughputs S can be expressed as:

$$\begin{aligned} S &\equiv \frac{\text{Average payload transmitted in a slot time}}{\text{Average length of a slot time}} = S_1 + S_2 \\ &= \frac{Q(1,0) \cdot E[P_{Len,1}] + Q(0,1) \cdot E[P_{Len,2}]}{\left\{ Q(0,0) \cdot \sigma + Q(1,0) \cdot T_{s,1} + Q(0,1) \cdot T_{s,2} + \sum_{0 \leq c_1 \leq n_1, 0 \leq c_2 \leq n_2, c_1 + c_2 \geq 2} Q(c_1, c_2) \cdot T_c(c_1, c_2) \right\}} \quad (5) \\ &\equiv \frac{Q(1,0) \cdot E[P_{Len,1}] + Q(0,1) \cdot E[P_{Len,2}]}{Q(0,0) \cdot \sigma + Q(1,0) \cdot T_{s,1} + Q(0,1) \cdot T_{s,2} + [1 - Q(0,0) - Q(1,0) - Q(0,1)] \cdot T_c} \end{aligned}$$

where S_1 and S_2 denote the throughputs contributed by type-1 and type-2 traffic flows, respectively. $E[P_{Len,i}]$ is the average duration to transmit the payload for type- i traffic (the payload size is measured with the time required to transmit it). For simplicity, with the assumption that all packets of type- i traffic have the same fixed size, we have $E[P_{Len,i}] = P_{Len,i} \cdot \sigma$ is the duration of an empty time slot. $T_{s,i}$ is the average time of a slot because of a successful transmission of a packet of a type- i traffic flow. $T_{s,i}$ can be expressed as

$$T_{s,i} = PHY_{header} + MAC_{header} + E[P_{Len,i}] + SIF + \delta + ACK + DIFS + \delta \quad (6)$$

where δ is the propagation delay. $T_c(c_1, c_2)$ is the average time the channel is sensed busy by each station during a collision caused by simultaneous transmissions of c_1 type-1 stations and c_2 type-2 stations. It can be expressed as

$$T_c(c_1, c_2) = PHY_{header} + MAC_{header} + \max[\theta(c_1)P_{Len,1}, \theta(c_2)P_{Len,2}] + DIFS + \delta \quad (7)$$

where

$$\theta(x) \equiv \begin{cases} 1 & x > 0 \\ 0 & x = 0 \end{cases}$$

Moreover, from equation (3), we can easily derive

$$(1 - p_1)(1 - \tau_1) = (1 - p_2)(1 - \tau_2) = \prod_{j=1}^2 (1 - \tau_j)^{n_j} \quad (8)$$

When the minimum contention window size $W_1 \gg 1$ and $W_2 \gg 1$, the transmission probabilities τ_1 and τ_2 are small, that is, $\tau_1 \ll 1$ and $\tau_2 \ll 1$. Therefore, from equation (8), we have the following approximation

$$p_1 \approx p_2 \quad (9)$$

When $W_1 \gg 1$, $W_2 \gg 1$ and $m_1 \approx m_2$, we have the following approximation based on equation (2)

$$(\tau_1 / \tau_2) \approx (W_2 / W_1) \quad (10)$$

From equations (4), (5) and (10), we finally have

$$\frac{s_1}{s_2} \equiv \frac{S_1 / n_1}{S_2 / n_2} = \frac{\frac{\tau_1}{1 - \tau_1} \cdot E[P_{Len,1}]}{\frac{\tau_2}{1 - \tau_2} \cdot E[P_{Len,2}]} \approx \left(\frac{E[P_{Len,1}]}{W_1} \right) / \left(\frac{E[P_{Len,2}]}{W_2} \right) \quad (11)$$

3 Maximum Throughput Analysis

We are interested in maximizing throughput, while *at the same time* ensuring service differentiation, and the hypothesis in this section is that differentiation is achieved by allocating bandwidth to the individual traffic flow to satisfy a given target ratio $\hat{\alpha} = s_2 / s_1$. For convenience, it is useful to define an additional *differentiation*

parameter $\alpha \equiv \left(\frac{\tau_2}{1 - \tau_2} \right) / \left(\frac{\tau_1}{1 - \tau_1} \right)$. According to equation (11), we have

$\alpha = \hat{\alpha} \cdot \frac{E[P_{Len,2}]}{E[P_{Len,1}]}$. In the following we always assume that the probabilities of transmission in a randomly selected slot time satisfy the constraints $0 \leq \tau_1 < 1$, $0 \leq \tau_2 < 1$.

Theorem 1: Assume that two types of traffic coexist in the system, with n_1 and n_2 numbers of traffic flows, respectively. If one fixes the desired differentiation:

$\frac{\tau_2}{1 - \tau_2} = \alpha \cdot \frac{\tau_1}{1 - \tau_1}$ ($\alpha > 0$), the throughput function $S(\tau_1, \tau_2)$ defined in equation (5)

has one and only one optimal operation point $\tau_1^*(\alpha)$ where the maximum throughput is achieved.

Proof:

From equation (5), we have

$$\begin{aligned}
 S(\tau_1, \tau_2) = & \frac{n_1 \frac{\tau_1}{(1-\tau_1)} E[P_{Len,1}] + \alpha n_2 \frac{\tau_1}{(1-\tau_1)} E[P_{Len,2}]}{\left\{ \begin{aligned} & \sigma + n_1 \frac{\tau_1}{1-\tau_1} \cdot T_{s,1} + \alpha n_2 \frac{\tau_1}{1-\tau_1} \cdot T_{s,2} \\ & + \alpha n_1 n_2 \left(\frac{\tau_1}{1-\tau_1} \right)^2 \cdot T_c(1,1) + n_1(n_1-1) \left(\frac{\tau_1}{1-\tau_1} \right)^2 \cdot T_c(2,0) \\ & + \alpha^2 n_2(n_2-1) \left(\frac{\tau_1}{1-\tau_1} \right)^2 \cdot T_c(0,2) + \frac{\sum_{i=1}^{n_1+n_2} Q(c_1, c_2) T_c(c_1, c_2)}{(1-\tau_1)^{n_1} (1-\tau_2)^{n_2}} \end{aligned} \right\}} \\
 = & (F_1 \cdot \chi) / \left(\sigma + \sum_{i=1}^{n_1+n_2} G_i \cdot \chi^i \right) \equiv \frac{F(\chi)}{G(\chi)}
 \end{aligned} \tag{12}$$

where $\chi \equiv \frac{\tau_1}{1-\tau_1}$ ($0 \leq \chi < +\infty$), F_1 and G_i ($i = 1, 2, \dots, n_1 + n_2$) are constants larger than zero. To determine the optimal operation point, we study the function:

$$\left(\frac{F(\chi)}{G(\chi)} \right)' = \frac{F(\chi)'G(\chi) - F(\chi)G(\chi)'}{G(\chi)^2} = \left(F_1\sigma - F_1 \sum_{i=2}^{n_1+n_2} (i-1)G_i\chi^i \right) / G(\chi)^2 \tag{13}$$

The optimal solution χ^* satisfies the following equation:

$$\sum_{i=2}^{n_1+n_2} (i-1)G_i(\chi^*)^i = \sigma \tag{14}$$

Because $\sigma > 0$ and $\sum_{i=2}^{n_1+n_2} (i-1)G_i\chi^i$ is a monotone increasing function with values ranging from 0 to $+\infty$ when χ varies from 0 to $+\infty$, the optimal χ^* must exist and be unique. From equation (13), it can be seen that $(F(\chi)/G(\chi))' > 0$ when $\chi < \chi^*$ and $(F(\chi)/G(\chi))' < 0$ when $\chi > \chi^*$. Therefore, the throughput function reaches the maximum value when $\frac{\tau_1^*}{1-\tau_1^*} = \chi^*$. Of course the optimal solution varies with the variation of the differentiation constant α . Therefore, we denote the optimal solution as $\tau_1^*(\alpha)$. \square

By using equation (14), the optimal operation point can be obtained by using a numerical method. However, in order to obtain a much deeper insight into the system performance, it is useful to derive more meaningful and concise approximations of the exact formulas. From equations (12) and (14), we have

$$\frac{\tau_1^*(\alpha)}{1-\tau_1^*(\alpha)} \leq \sqrt{\frac{\sigma}{G_2}} = \sqrt{\frac{\sigma}{\alpha n_1 n_2 T_c(1,1) + n_1(n_1-1)T_c(2,0) + \alpha^2 n_2(n_2-1)T_c(0,2)}} \quad (15)$$

It can be seen that, if n_1 , n_2 , $E[P_{Len,1}]$ and $E[P_{Len,2}]$ are sufficiently large, the optimal operation point $\tau_1^*(\alpha)$ is far less than one (it is also true for $\tau_2^*(\alpha)$). Therefore, it is reasonable to limit the discussions to the case that $\tau_1 \ll 1$ and $\tau_2 \ll 1$.

Theorem 2: Assume that two types of traffic coexist in the system with n_1 and n_2 flows, respectively. Moreover, assume that $\frac{\tau_2}{1-\tau_2} = \alpha \cdot \frac{\tau_1}{1-\tau_1}$ ($\alpha > 0$). If n_1 , n_2 , $E[P_{Len,1}]$ and $E[P_{Len,2}]$ are sufficiently large so that the optimal operation point $\tau_1^*(\alpha) \ll 1$, $\tau_2^*(\alpha) \ll 1$, then the optimal operation point can be approximated as

$$\tau_1^*(\alpha) \approx 1 / \left((n_1 + \alpha n_2) \sqrt{T_c^*/2} \right) \equiv \tau_{1_ap}^*(\alpha) \quad (16)$$

where $T_c^* \equiv T_c / \sigma$. Moreover, if $E[P_{Len,1}] = E[P_{Len,2}] = P_{Len}$, the corresponding achieved maximum throughput can be approximated as

$$S_{\max} \approx P_{Len} / \left(T_s + \sigma K + T_c [K(e^{1/K} - 1) - 1] \right) \quad (17)$$

where $K \equiv \sqrt{T_c^*/2}$.

Proof:

According to Theorem 1, because at the optimal operation point $\tau_1^*(\alpha) \ll 1$, $\tau_2^*(\alpha) \ll 1$, we can limit our discussion only to the range of $\tau_1 \ll 1$, $\tau_2 \ll 1$. In this case, the relationship $\frac{\tau_2}{1-\tau_2} = \alpha \cdot \frac{\tau_1}{1-\tau_1}$ ($\alpha > 0$) can be approximated as $\tau_2 \approx \alpha \cdot \tau_1$.

First, if we neglect the case that three or more packets collide with each other at the same time, we have

$$\begin{aligned} T_c &\approx \frac{Q(1,1) \cdot T_c(1,1) + Q(2,0) \cdot T_c(2,0) + Q(0,2) \cdot T_c(0,2)}{Q(1,1) + Q(2,0) + Q(0,2)} \\ &\approx \frac{\alpha n_1 n_2 \cdot T_c(1,1) + n_1(n_1-1) \cdot T_c(2,0) + \alpha^2 n_2(n_2-1) \cdot T_c(0,2)}{\alpha n_1 n_2 + n_1(n_1-1) + \alpha^2 n_2(n_2-1)} \end{aligned} \quad (18)$$

From the above approximation, it can be seen that once $E[P_{Len,1}]$, $E[P_{Len,2}]$, n_1 , n_2 and α are given, T_c can be regarded as a constant.

Based on the assumption that $\tau_1 \ll 1$, $\tau_2 \ll 1$ and on equation (8), equation (5) can be approximated as follows:

$$\begin{aligned}
S(\tau_1, \tau_2) &\approx \frac{n_1 \tau_1 (1-p_1) E[P_{Len,1}] + n_2 \alpha \tau_1 (1-p_1) E[P_{Len,2}]}{\left\{ (1-\tau_1)(1-p_1) \sigma + n_1 \tau_1 (1-p_1) T_{s,1} + n_2 \alpha \tau_1 (1-p_1) T_{s,2} \right\}} \\
&\quad \left\{ + [1 - (1-\tau_1)(1-p_1) - n_1 \tau_1 (1-p_1) - n_2 \alpha \tau_1 (1-p_1)] T_c \right\} \\
&\approx \frac{n_1 \tau_1 E[P_{Len,1}] + n_2 \alpha \tau_1 E[P_{Len,2}]}{\sigma + n_1 \tau_1 T_{s,1} + n_2 \alpha \tau_1 T_{s,2} + [(1-p_1)^{-1} - 1 - n_1 \tau_1 - n_2 \alpha \tau_1] T_c} \equiv \frac{f(\tau_1)}{g(\tau_1)}
\end{aligned} \tag{19}$$

Approximately, the optimal solution must satisfy the following condition

$$f(\tau_1^*) / f'(\tau_1^*) = g(\tau_1^*) / g'(\tau_1^*) \tag{20}$$

That is,

$$\tau_1^* = \frac{\sigma + n_1 \tau_1^* T_{s,1} + n_2 \alpha \tau_1^* T_{s,2} + \left[\frac{1}{1-p_1} - 1 - n_1 \tau_1^* - n_2 \alpha \tau_1^* \right] \cdot T_c}{n_1 T_{s,1} + n_2 \alpha T_{s,2} + [d(1-p_1)^{-1} / d\tau_1]_{\tau_1=\tau_1^*} - n_1 - n_2 \alpha} \cdot T_c \tag{21}$$

After some simplifications of the above equation, one obtains

$$(n_1 + \alpha n_2) \tau_1^* T_c^* = (1-p_1) \Big|_{\tau_1=\tau_1^*} \cdot (1-T_c^*) + T_c^* \approx (1-\tau_1^*)^{n_1} (1-\alpha \tau_1^*)^{n_2} (1-T_c^*) + T_c^* \tag{22}$$

Because $(1-\alpha \tau_1^*)^{n_2} \approx 1 - \alpha n_2 \tau_1^* \approx (1-\tau_1^*)^{\alpha n_2}$, the above equation can be further approximated as

$$(n_1 + \alpha n_2) \tau_1^* T_c^* = (1-\tau_1^*)^{n_1 + \alpha n_2} \cdot (1-T_c^*) + T_c^* \tag{23}$$

When there is only one type of traffic, equation (23) is actually the same as equation (27) in [10]. By referring to equation (28) in [10], equation (16) can be obtained.

Next, we evaluate the maximum throughput by substituting the approximate optimal solution $\tau_{1-ap}^*(\alpha)$ into equation (5).

$$\begin{aligned}
S_{\max} &\approx \frac{n_1 \tau_{1-ap}^* (1-\tau_{1-ap}^*)^{n_1 + \alpha n_2} E[P_{Len,1}] + \alpha n_2 \tau_{1-ap}^* (1-\tau_{1-ap}^*)^{n_1 + \alpha n_2} E[P_{Len,2}]}{\left\{ (1-\tau_{1-ap}^*)^{n_1 + \alpha n_2} \sigma + n_1 \tau_{1-ap}^* (1-\tau_{1-ap}^*)^{n_1 + \alpha n_2} T_{s,1} \right.} \\
&\quad \left\{ + \alpha n_2 \tau_{1-ap}^* (1-\tau_{1-ap}^*)^{n_1 + \alpha n_2} T_{s,2} + [1 - (1-\tau_{1-ap}^*)^{n_1 + \alpha n_2} \right.} \\
&\quad \left. \left. - n_1 \tau_{1-ap}^* (1-\tau_{1-ap}^*)^{n_1 + \alpha n_2} - \alpha n_2 \tau_{1-ap}^* (1-\tau_{1-ap}^*)^{n_1 + \alpha n_2} \right] \cdot T_c \right\}
\end{aligned} \tag{24}$$

Because n_1 and n_2 are assumed sufficiently large, we have the following approximation:

$$\left(1 - ((n_1 + \alpha n_2) K)^{-1} \right)^{n_1 + \alpha n_2} \approx e^{-1/K} \tag{25}$$

Moreover, we assume $E[P_{Len,1}] = E[P_{Len,2}] = P_{Len}$ and therefore $T_{s,1} = T_{s,2} = T_s$, then equation (24) can be further approximated as equation (17). \square

Deduction 1: Assume that $L \geq 1$ types of traffic coexist in the system, with numbers of type- i traffic flows n_i ($i=1,2,\dots,L$). Moreover, assume that $\frac{\tau_i}{1-\tau_i} = \alpha_i \cdot \frac{\tau_1}{1-\tau_1}$ ($\alpha_i > 0, i=1,2,\dots,L, \alpha_1 \equiv 1$). If n_i ($i=1,2,\dots,L$) and $E[P_{Len,i}]$ ($i=1,2,\dots,L$) are sufficiently large so that the optimal operation point $\tau_i^*(\alpha_1, \dots, \alpha_L) \ll 1$ ($i=1,2,\dots,L$), then the optimal operation point can be approximated as

$$\tau_1^*(\alpha_2, \dots, \alpha_L) \approx 1 / \left(\sum_{j=1}^L \alpha_j n_j \sqrt{\frac{T_c^*}{2}} \right) \equiv \tau_{1_ap}^*(\alpha_2, \dots, \alpha_L) \quad (26)$$

where $T_c^* \equiv T_c / \sigma$. Moreover, if $E[P_{Len,1}] = \dots = E[P_{Len,L}] = P_{Len}$, the corresponding achieved maximum throughput can be approximated as

$$S_{\max} \approx P_{Len} / (T_s + \sigma K + T_c [K(e^{1/K} - 1) - 1]) \quad (27)$$

where $K \equiv \sqrt{T_c^* / 2}$. \square

Compared with the equation (31) in [10], we find that *the maximum throughput achieved is exactly the same no matter how many different types of traffic flows coexisting in the system.*

Deduction 2: Assume that there are $L \geq 1$ types of traffic coexisting in the system with n_i ($i = 1, 2, \dots, L$) traffic flows. Moreover, assume that $\frac{\tau_i}{1 - \tau_i} = \alpha_i \cdot \frac{\tau_1}{1 - \tau_1}$ ($\alpha_i > 0, i = 1, 2, \dots, L, \alpha_1 \equiv 1$). If n_i ($i = 1, 2, \dots, L$) and $E[P_{Len,i}]$ ($i = 1, 2, \dots, L$) are sufficiently large so that the optimal operation point $\tau_i^*(\alpha_1, \dots, \alpha_L) \ll 1$ ($i = 1, 2, \dots, L$), then the system operates close to the optimal operation point if and only if the packet collision rate is approximately equal to $1 - e^{-1/K}$ ($K \equiv \sqrt{T_c^* / 2}$). \square

The above equation can be used to check if the system works close to the optimal operation point.

4 Validation of Approximations

In this section, we validate the approximated results obtained in the former section by using a numerical method. The parameters for the system are summarized in Table 1, based on IEEE 802.11b.

In the first example, we compare the exact optimal operation points τ_1^* numerically obtained from equation (5) with the approximated optimal operation

Table 1. System Parameters

MAC Header	272 bits
PHY Header	192 μ s
ACK	112 bits + PHY header
Channel Bit Rate	11Mbps
Propagation Delay	1 μ s
Slot Time	20 μ s
SIFS	10 μ s
DIFS	50 μ s

points $\tau_{1_ap}^*$ obtained from equation (16). In the example, we set other parameters as:

$$\frac{\tau_2}{1-\tau_2} = \alpha \frac{\tau_1}{1-\tau_1}, \quad \frac{n_2}{n_1} = 2, \quad P_{Len,1} = P_{Len,2} = 2000 \text{ bytes}, \text{ and } m_1 = m_2 = 8.$$

In Fig. 2, the comparison results of optimal operation points are shown versus the number of type-1 traffic flows n_1 . Two cases are shown in the figure: one is for the case that $\alpha = 0.1$ and the other is $\alpha = 10$. From the figure, it can be seen that good agreements between exact and approximate optimal operation points can be achieved if the number of traffic flows n_1 is not so small. Furthermore, comparisons between the case of $\alpha = 0.1$ and that of $\alpha = 10$ show that good estimation accuracy can be obtained as long as the estimated optimal operation point are far less than one.

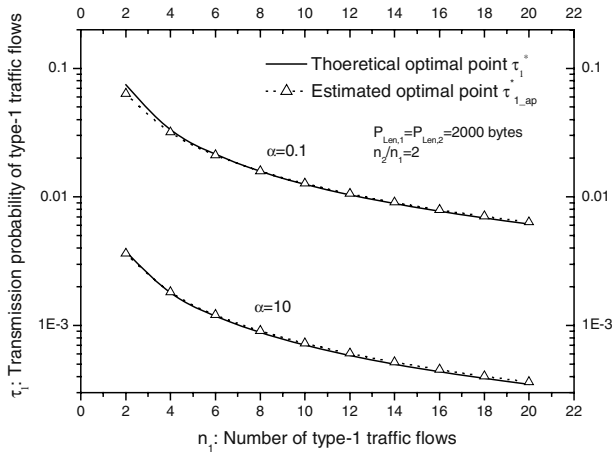


Fig. 2. Comparisons between theoretical optimal operation points and estimated ones

After verifying the accuracy of the estimation for the optimal operation point, we illustrate the accuracy of the evaluated maximum throughput by using the estimated optimal operation point. In order to obtain the exact maximum throughput and its evaluated value, we substitute exact optimal operational point and its corresponding approximated one into equation (5) respectively. The comparison results are given in Table 2. From the Table, it can be seen that the estimated maximum throughput S_{max_ap} accord with the corresponding theoretical value S_{max} very well. Moreover, in the Table, we show the evaluated maximum throughput obtained from equation (17). It can be regarded as the limiting value for the maximum throughput when $n_1 \rightarrow \infty$.

5 An Adaptive Scheme to Achieve Maximum Throughput and Service Differentiation

For the implementation of real-world systems, in addition to the existence of an optimal operation point, one is interested in methods to reach the point and to maintain a dynamic system close to the optimal point. In the following part, we present two schemes for this purpose.

Table 2. Comparisons between theoretical maximum throughput and estimated ones

n ₁	$\alpha=0.1$		$\alpha=10$	
	S _{max}	S _{max_ap}	S _{max}	S _{max_ap}
6	0.66521	0.66518	0.66323	0.66322
8	0.66383	0.66381	0.66237	0.66235
10	0.66301	0.66299	0.66187	0.66183
12	0.66248	0.66245	0.66153	0.66148
14	0.66210	0.66206	0.66129	0.66123
16	0.66181	0.66177	0.66111	0.66105
18	0.66159	0.66155	0.66097	0.66091
20	0.66142	0.66137	0.66086	0.66079
∞	0.65976			

System parameters: $P_{Len,1} = P_{Len,2} = 2000$ bytes, $n_2 = 2n_1$, $m_1 = m_2 = 8$

5.1 Basic Adaptive Scheme

Based on equation (11), to achieve a certain target service differentiation $\hat{\alpha} = s_2/s_1$,

we can adjust the ratio $\left(\frac{\tau_2}{1-\tau_2}\right) \bigg/ \left(\frac{\tau_1}{1-\tau_1}\right)$ to be $\alpha = \hat{\alpha} \cdot \frac{E[P_{Len,1}]}{E[P_{Len,2}]}$. In this case, if the

optimal operation point $\tau_1^* \ll 1$ and $\tau_2^* \ll 1$, from Theorem 2, they can be approximated as $\tau_1^* \approx 1/\left((n_1 + \alpha n_2) \cdot \sqrt{T_c^*/2}\right)$ and $\tau_2^* \approx 1/\left(\left(\frac{n_1}{\alpha} + n_2\right) \cdot \sqrt{T_c^*/2}\right)$,

respectively. Therefore, if $E_1 \equiv n_1 + \alpha n_2$ and $E_2 \equiv \frac{n_1}{\alpha} + n_2 = \frac{E_1}{\alpha}$ are known and the

packet transmission probabilities of traffic flows are equal to $\tau_1^* \approx 1/\left(E_1 \cdot \sqrt{T_c^*/2}\right)$

and $\tau_2^* \approx 1/\left(E_2 \cdot \sqrt{T_c^*/2}\right)$ respectively, the system operates *almost* at the optimal

point and the service differentiation achieved can be approximated as

$$\frac{s_2}{s_1} \approx \hat{\alpha} = \alpha \cdot \frac{E[P_{Len,2}]}{E[P_{Len,1}]}.$$

Assuming that n_1 , n_2 and α are known, the problem is how to make packet transmission probabilities τ_1 and τ_2 reach their corresponding approximate optimal values $\tau_{1_ap}^*$ and $\tau_{2_ap}^*$. First, each station can evaluate the average frame collision length T_c^* at run-time. Next, it calculates the target optimal packet transmission probabilities $\tau_{1_ap}^*$ or $\tau_{2_ap}^*$ based on Theorem 2, and the approximate packet collision rate $p_{_ap}^*$ corresponding to the optimal operation point by using Deduction 2. Then, by substituting $\tau_{1_ap}^*$, $\tau_{2_ap}^*$ and $p_{_ap}^*$ into equation (2), one can obtain the approximate optimal minimum contention window size $W_{1_ap}^*$ and $W_{2_ap}^*$. Finally, $W_{1_ap}^*$ and $W_{2_ap}^*$ are used to adjust the current minimum contention window size $Current_W_1$ and $Current_W_2$ as follows:

$$Current_W_i = \beta \cdot Current_W_i + (1 - \beta) \cdot W_{i_ap}^* \tag{28}$$

where $i = 1, 2$, and $\beta \in [0, 1]$ is a smoothing factor, which determines the convergence speed of the scheme.

We simulated the above scheme to verify its performance. In the simulation, it is assumed that $n_1 = 10$, $n_2 = 20$ are known. In this case, no central controller is needed. Parameter α is set as 0.2. The frame lengths of both traffic types are equal. Both traffic flows begin their minimum contention window size from 512.

Table 3 shows the comparison between the theoretical maximum throughput S_{max} and the actual throughput S and the service differentiation s_1/s_2 achieved by using the basic adaptive scheme. It can be seen that the proposed adaptive scheme can achieve the maximum throughput and at the same time the target service differentiation performance.

Table 3. Comparisons between theoretical maximum throughput and simulated ones

P _{Len} (bytes)	S _{max}	S	s ₁ /s ₂
500	0.36199	0.36235	5.01728
700	0.43628	0.43588	5.02386
900	0.49298	0.49316	5.07283
1100	0.53786	0.53677	4.98651
1300	0.57437	0.57460	5.05422
1500	0.60471	0.60646	5.02912
1700	0.63038	0.63139	4.97099
1900	0.65241	0.65302	5.10741
2100	0.67155	0.67105	5.11010

$$n_1=10, n_2=20, 1/\alpha=5.0, \beta=0.8, m_1=m_2=8$$

In the basic adaptive scheme, it is assumed that n_1 , n_2 and α are known (hence E_1 and E_2 are known). The optimal operation point $\tau_{1_ap}^*$, $\tau_{2_ap}^*$ are mainly

determined by the value of E_1 and E_2 . However, extensive simulations show that the sensitivity of the achieved throughput to changes of E_1 is small, when the differentiation parameter α is fixed. To some extent, the system can achieve optimal performance by using the basic adaptive scheme even the actual number of traffic flows are different from the assumed ones. This is because the throughput function in equation (5) is very smooth with the variation of τ_1 . However, for large deviations of E_1 from the assumed value, the achieved throughput deteriorates.

5.2 A Centralized Adaptive Scheme

A centralized version of the adaptive scheme uses a central controller (CC) is proposed in this section. Let us note that a centralized network control can be assumed in a *hot spot* scenario, with the need of identifying users, accounting and billing, managing and supporting QoS, possibly also through pricing and call admission control (CAC) [27]. In our scheme, the CC itself carries traffic flows for transmission (we assume of type-1) and, in addition, it serves as a coordinator to guarantee that the centralized knowledge can be used to achieve the maximum throughput and target service differentiation even in a dynamic context, when the number of active mobile stations changes. The functions of a CC in the improved scheme can be explained as follows: It detects the value of E_1 and E_2 at run time. If the detected value of E_1 and E_2 are sufficiently far from the current estimates, the CC broadcasts the new estimates. In order to maintain the target service differentiation between different traffic flows, the CC also broadcasts the target differentiation ratio. After receiving the new values, all mobile terminals in the current basic service set (BSS) modify their memorized values of E_1 and E_2 and use the adaptive scheme described previously.

To keep track of the number of active mobile stations, the CC monitors the traffic and evaluates the real-time values of E_1 and E_2 as follows. In the case that $\tau_1 \ll 1$, $\tau_2 \ll 1$, $\tau_2 = \alpha\tau_1$ and by using equation (8), one has

$$(1 - p_1) \approx (1 - \tau_1)^{n_1 + \alpha n_2} = (1 - \tau_1)^{E_1} \quad (29)$$

From above equation, one estimates E_1 as

$$\hat{E}_1 = \log(1 - p_1) / \log(1 - \tau_1) \quad (30)$$

where the packet collision rate p_1 can be easily evaluated at run-time. An efficient way to evaluate the run-time packet collision rate is proposed in [28]. τ_1 is obtained by substituting the estimated p_1 and the current minimum contention window size $Current_W_1$ into equation (2). After obtaining \hat{E}_1 , it is averaged as \bar{E}_1 and compared with the $Current_E_1$, which is the current memorized value for E_1 . \bar{E}_1 can be expressed as

$$\bar{E}_1 = \beta \cdot \bar{E}_1 + (1 - \beta) \cdot \hat{E}_1 \quad (31)$$

Table 4. Performance of the modified adaptive scheme

n_1, n_2	S_{\max}	S	s_1/s_2
2,4	0.67338	0.66721	5.67252
5,10	0.66486	0.66508	5.35558
10,20	0.66230	0.66184	4.96943
20,40	0.66107	0.66238	5.00814
30,60	0.66066	0.65910	4.93129
50,100	0.66035	0.65292	4.83726

$P_{\text{Len}}=2000$ bytes, $1/\gamma = 5.0$, $\gamma = 0.8$, $m_1=m_2=8$

If \bar{E}_1 is less than $\text{Current_}E_1 \cdot \gamma$ ($0 < \gamma < 1$) during the past $k_t \geq 1$ comparisons, the $\text{Current_}E_1$ will be set as \bar{E}_1 . If \bar{E}_1 is larger than $\text{Current_}E_1 / \gamma$ ($0 < \gamma < 1$) during the past $k_t \geq 1$ comparisons, the $\text{Current_}E_1$ will be set as \bar{E}_1 . $\text{Current_}E_2$ is simply obtained as $\text{Current_}E_1 / \alpha$. In the scheme, if γ is set to be 0, the improved scheme is actually the same as the basic scheme. On the other hand, if γ is very close to 1, the CC will modify E_1 and E_2 too often, which proves to be unnecessary according to the former discussions about the sensitivities of achieved throughput to the number of traffic flows. Therefore, parameters γ and k_t should be carefully chosen to improve the performance of the system and to minimize the control overhead.

The performance of the improved scheme is verified by simulation. In the simulation, a station carrying type-1 traffic flow serves as the CC. γ and k_t are set to be 0.5 and 10, respectively. If the CC decides to broadcast new values for E_1 and E_2 , it generates a special management frame and gains access to the channel by using the highest medium access priority (PIFS) to ensure the new values can be received as soon as possible. Table 4 shows the performance of the centralized adaptive scheme. We can see that the achieved throughput S is now close to the corresponding maximum throughput S_{\max} for all the cases, which is caused by the ability to adapt to dynamically changing values of E_1 and E_2 . Moreover, that service differentiation ratio s_1 / s_2 is kept approximately constant.

6 Conclusions

In this paper, we use a model of a wireless LAN based on the *standard* IEEE 802.11 MAC with a simple extension for service differentiation and derive approximations to get simpler but more meaningful relationships among the different parameters. We successfully derive the best operation point where the maximum throughput can be achieved and demonstrate its uniqueness. In addition we propose simple rules to decide if the system works under the optimal state. The other contribution of the paper is the proposal of two adaptive schemes (one distributed and the other one centralized) to lead and maintain the system close to the optimal operation point while

at the same time guaranteeing target service differentiation between different traffic types.

References

1. Y. Cheng and W. H. Zhuang, "DiffServ resource allocation for fast handoff in wireless mobile Internet," *IEEE Communications Magazine*, vol. 40, no. 5, 2002, pp. 130–136.
2. R. Braden, D. Clark and S. Shenker, "Integrated services in the Internet architecture: an overview," *IETF RFC 1633*, Jun. 1994.
3. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for differential services," *IETF RFC 2475*, Dec. 1998.
4. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, *IEEE Standard 802.11*, Aug. 1999.
5. J. Weinmiller, M. Schlager, A. Festag, and A. Wolisz, "Performance study of access control in wireless LANs *IEEE 802.11 DFWMAC* and *ETSI RES 10 HIPERLAN*," *Mobile Networks and Applications*, vol. 2, pp. 55–67, 1997.
6. H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of *IEEE 802.11 MAC* protocol," *Wireless Networks*, vol. 3, pp. 217–234, 1997.
7. T. S. Ho and K. C. Chen, "Performance evaluation and enhancement of the CSMA/CA MAC protocol for *802.11* wireless LAN's," *Proceedings of IEEE PIMRC*, Taipei, Taiwan, Oct. 1996, pp. 392–396.
8. F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 wireless LAN: Capacity analysis and protocol enhancement," *Proceedings of INFOCOM'98*, San Francisco, CA, March 1998, vol. 1, pp. 142–149.
9. G. Bianchi, L. Fratta, and M. Oliveri, "Performance analysis of *IEEE 802.11 CSMA/CA* medium access control protocol," *Proceedings of IEEE PIMRC*, Taipei, Taiwan, Oct. 1996, pp. 407–411.
10. G. Bianchi, "Performance analysis of the *IEEE 802.11* distributed coordination function," *IEEE Journal on Selected Areas In Communications*, vol. 18, no. 3, March 2000.
11. Y. C. Tay and K. C. Chua, "A Capacity Analysis for the *IEEE 802.11 MAC* Protocol," *Wireless Networks*, 7, 2001, pp. 159–171.
12. J. L. Sobrinho and A. S. Krishnakumar, "Distributed multiple access procedures to provide voice communications over *IEEE 802.11* wireless networks," *Proceedings GLOBECOM 1996*, pp. 1689–1694.
13. J. Deng and R.S. Chang, "A priority scheme for *IEEE 802.11 DCF* access method," *IEICE Transactions in Communications*, vol. 82-B, no. 1, Jan 1999, pp. 96–102.
14. A. Veres, A. T. Campbell, M. Barry and L. H. Sun, "Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control," *IEEE Journal on Selected Areas In Communications*, vol. 19, no. 10, Oct 2001, pp. 2081–2093.
15. I. Aad and C. Castelluccia, "Differentiation Mechanisms for *IEEE 802.11*," *Proceedings of IEEE Inforcom 2001*, pp. 209–218.
16. S. Mangold, S. Choi, P. May, O. Klein, G. Hietz and L. Stibor, "IEEE 802.11e wireless lan for quality of service," *Proceedings of the European Wireless*, Feb 2002.
17. Z. Jun, G. Zihua, Z. Qian and Z. Wenwu, "Performance Study of MAC for Service Differentiation in *IEEE 802.11*," *Proceedings of the GLOBECOM '02*, IEEE, Volume: 1, Nov 17-21, 2002 pp. 778–782.
18. Bo LI, Roberto Battiti, "Supporting Service Differentiation with Enhancements of the *IEEE 802.11 MAC* Protocol: Models and Analysis" Technical Report of Department of Computer Science and Telecommunications of University of Trento, no. DIT-03-024, available at <http://dit.unitn.it/research/publications/techRep?id=418>

19. F. Cali, M. Conti and E. Gregori "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Transactions on Networking*, vol. 8, No. 6, Dec 2000, pp. 785–799.
20. L. Romdhani, Q. Ni, and T. Turletti, "AEDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks," INRIA Technical Report. <http://www.inria.fr/rrrt/rr-4544.html>
21. T. Ozugur, M. Naghshineh, P. Kermani, C. Michael, B. Rezvani and J. A. Copeland, "Balanced media access methods for wireless networks," in *Proc. ACM MobiCom'98*, Dallas, TX, Oct. 1998, pp.21–32.
22. N. H. Vaidya, P. Bahl and S. Gupta, "Distributed fair scheduling in a wireless Lan," *ACM Mobicom'2000*. <http://research.microsoft.com/users/bahl/papers/pdf/mobiCom2000.pdf>
23. D. Qiao and K. G. Shin, "Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF," *Quality of Service*, 2002. Tenth IEEE International Workshop, 2002, Page(s): 227–236
24. M. Benveniste, G. Chesson, M. Hoehen, A. Singla, H. Teunissen, and M. Wentink, "EDCF proposed draft text," IEEE working document 802.11-01/131r1, March 2001.
25. A. Lindgren, A. Almquist, and O. Schelen, "Evaluation of quality of service schemes for IEEE 802.11 wireless LANs," *Proceedings of IEEE Conference on Local Computer Networks (LCN 2001)*, November 15–16, 2001, pp. 348–351.
26. G. Bolch, S. Greiner, H. de Meer, and K.S. Trivedi, *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, Wiley-Interscience, 1998, pp. 140–144.
27. Roberto Battiti, Marco Conti, Enrico Gregori, Mikalai Sabel, "Price-based Congestion-Control in Wi-Fi Hot Spots," *Proceedings of WiOpt'03* March 3–5, 2003, INRIA Sophia-Antipolis, France, pp. 91–100.
28. G. Bianchi and I. Tinnirello, "Kalman Filter Estimation of the number of Competing Terminals in an IEEE 802.11 Network," *IEEE INFOCOM 2003*.
29. Bo Li, Roberto Battiti, "Performance Analysis of An Enhanced IEEE 802.11 Distributed Coordination Function Supporting Service Differentiation," *QoFIS (International Workshop on Quality of Future Internet Services) 2003*, Sweden, Springer Lecture Notes on Computer Science LNCS volume 2811, pp. 152–161.

Throughput of the Multi-hop Slotted Aloha with Multi-packet Reception

M. Coupechoux^{1,2}, T. Lestable^{1,3}, C. Bonnet², and V. Kumar¹

¹ Alcatel Research & Innovation, route de Nozay, 91460 Marcoussis, France,
marceau.coupechoux@alcatel.fr,

² Institut Eurecom, Mobile Communications Dpt., Sophia-Antipolis, France,

³ Supélec, Radio Dpt., Gif-sur-Yvette, France

Abstract. In this paper, we analyse the throughput of a multihop network, where nodes use slotted ALOHA as medium access protocol (MAC) and are able to receive simultaneously several packets in a slot. We provide a closed-form formula for the throughput in the generic case as a function of the probability, $r_{n,k}$, for a receiver to decode k packets given that n have been sent in its neighborhood. We then consider several simple models for the computation of the $r_{n,k}$, when spread slotted ALOHA is used. In particular, we compare the performances of a matched filter (MF) receiver with those of a linear minimum mean-square error (MMSE) multi-user detector (MUD). Capacity results show the great advantage of multi-packet reception and highlight the near-far resistance of the MUD scheme.

1 Introduction

In recent years, a lot of effort has been spent in the design of protocols for ad hoc networks. Such packet networks are multi-hop and operate without any fixed infrastructure. This can be a low cost and easily deployable technology to provide high speed Internet access in a wireless environment, to organize networks of sensors, or to complement the coverage of future cellular networks.

In this paper, we pay special attention to the MAC sub-layer and, in particular, to the traditional slotted ALOHA scheme. A lot of protocols have been proposed in the literature to address the issue of medium access, and we can distinguish two main families: the contention based schemes and the conflict-free schemes. Slotted ALOHA is the most simple protocol of the first category. Other examples are MACA [1], MACAW [2], FAMA [3], or IEEE 802.11 DCF [4]. On the other hand, conflict-free protocols allow the reservation of the channel for a certain amount of time and transmissions are then conflict-free. In this case, the reservation phase or the transmission of broadcast packets often relies on a slotted ALOHA scheme. That is the reason why this simple protocol is of great interest.

The spatial capacity of the slotted ALOHA protocol has been studied in [5], where the effect of capture is detailed. This capacity has been obtained with the assumption that receivers devices are able to decode at most a single

packet per slot. However, research performed since the early 1980's in the domain of multi-user detection in CDMA systems [14] shows that this condition can be overcome. Indeed, receivers using multi-user detection schemes can decode the packets from several simultaneous transmitters. In particular, the near-far resistance of the multi-user detectors [13] makes this technique very attractive for ad hoc networks, where power control schemes are much more difficult to implement than in traditional single-hop systems.

In this paper, we extend the result of [5] in the case of multi-packet reception (section 2) and we provide a closed-form formula for the throughput of the slotted ALOHA as a function of the probability, $r_{n,k}$, for a receiver to decode k packets given that n have been sent in its neighborhood. In section 3, we detail three different models of multi-packet reception: a simple model often used in the literature, a bank of MF, and a MMSE multi-user detector. At last, in section 4, we provide numerical results and highlight the near-far resistance of the MUD scheme.

2 Spatial Throughput with Multi-packet Reception

2.1 Models

Throughout this paper, we will consider a packet radio network of nodes, spatially distributed in the plane according to a Poisson process with parameter λ . That means that the probability to find k nodes in any region, A , of area $S(A)$ is:

$$P[k \text{ in } A] = \frac{(\lambda S(A))^k}{k!} e^{-\lambda S(A)}. \quad (1)$$

We will assume that the considered network is large and we will neglect the edge effects.

All nodes are assumed to operate with a half-duplex radio device. This means that a collision of the second order can occur if a node receives a packet, while it is itself transmitting during the same slot. In this case, the packet is lost. The transmit power is constant and equals P_0 .

As explained in the introduction, we assume that nodes access the channel by using the slotted ALOHA protocol, i.e., time is divided in equal time-slots. At a given slot, a node sends a packet with a fixed probability p . Otherwise, it is able to receive one or several packets coming from the transmitters. Let R_0 be the reception radius of a receiver. R_0 is the maximum distance from which can come a packet destined to this receiver. If there are n transmitters within R_0 from the receiver, the probability to decode k packets is $r_{n,k}$.

We assume that packets destined towards a particular node in the network are routed with equal probability towards one of the neighboring nodes that lies in the direction of the destination. All these assumptions are taken from [5].

First of all, we are interested in the local throughput of the system, i.e., the expected number of packet received per slot. We will then evaluate the expected forward progress of a packet and conclude our study with the total throughput of the network.

2.2 Preliminary Results

Before looking at the local throughput, we recall two preliminary results already given in [5]. We consider a particular node a and we define the random variable X as the number of correctly decoded packets destined to a in a given slot. Let us define two important events: (A) the event that a does not transmit; (T) the event that a particular sender t sends a packet to a . We have the two basic results:

$$P[A] = 1 - p, \quad (2)$$

$$P[T] = \frac{1 - e^{-\lambda\pi R_0^2/2}}{\lambda\pi R_0^2}, \quad (3)$$

where p is the probability of transmission, λ is the density of the nodes, and R_0 is the transmission range. Note that if nodes are spatially distributed according to a Poisson process with density λ , senders, at a given time-slot, are spatially distributed according to a Poisson process with density λp (see e.g. [16]).

2.3 Local Throughput

We are now in position to evaluate the local throughput. Let us define two more events: (T_n) the event that there are n senders in the neighborhood of a ; (D_k) the event that a decodes exactly k packets in the given time-slot. Now, the probability that a receives x packets given (A) , (T_n) , and (D_k) is:

$$P[X = x|A, T_n, D_k] = \binom{k}{x} P[T]^x (1 - P[T])^{k-x}, k \geq x, \quad (4)$$

because among the k packets decoded, x are destined to a . This probability is zero if $k < x$. We now successively un-condition this relation:

$$P[X = x|A, T_n] = \sum_{k=0}^n P[X = x|A, T_n, D_k] P[D_k|A, T_n] \quad (5)$$

$$= \sum_{k=0}^n P[X = x|A, T_n, D_k] r_{n,k} \quad (6)$$

$$= \sum_{k=x}^n \binom{k}{x} P[T]^x (1 - P[T])^{k-x} r_{n,k}. \quad (7)$$

The second line is justified by the fact that the events (D_k) and (A) are independent. The third line takes into account Eq. 4. Now, assuming that the considered node a does not transmit:

$$P[X = x|A] = \sum_{n=0}^{\infty} P[X = x|A, T_n] P[T_n|A] \quad (8)$$

$$= \sum_{n=0}^{\infty} P[X = x|A, T_n] \frac{(\lambda p \pi R_0^2)^n}{n!} e^{-\lambda p \pi R_0^2} \quad (9)$$

$$= \sum_{n=0}^{\infty} \sum_{k=x}^n \binom{k}{x} P[T]^x (1 - P[T])^{k-x} r_{n,k} \frac{(\lambda p \pi R_0^2)^n}{n!} e^{-\lambda p \pi R_0^2} . \quad (10)$$

The second equation results from the fact that (T_n) and (A) are independent and that the density of the senders is λp as explained before. Note that if a is a sender at the considered slot, a cannot receive any packet because of the half-duplex nature of its radio device. So, for $x \neq 0$:

$$\begin{aligned} P[X = x] &= P[X = x|A]P[A] \\ &= P[X = x|A](1 - p) , \end{aligned} \quad (11)$$

according to Eq.2. We have obtained the probability distribution (pdf) function of X , the number of packets received by a :

$$P[X = x] = \sum_{n=0}^{\infty} \sum_{k=x}^n \binom{k}{x} P[T]^x (1 - P[T])^{k-x} r_{n,k} \frac{(\lambda p \pi R_0^2)^n}{n!} e^{-\lambda p \pi R_0^2} (1 - p) . \quad (12)$$

The throughput in a is immediatly obtained by taking the expectation of X :

$$E[X] = \sum_{x=1}^{\infty} x P[X = x] . \quad (13)$$

If there are N nodes in the network, the local throughput, S , of the network, i.e., the throughput at the MAC layer is:

$$S = N E[X] . \quad (14)$$

Note that the single-packet detection without capture is a special case of the aboves formulas. Indeed, by taking $r_{1,1} = 1$, $r_{n,0} = 1$ for $n \neq 1$, and $r_{n,k} = 0$ otherwise, we get:

$$\begin{aligned} E[X] &= P[X = 1] \\ &= P[T](\lambda p \pi R_0^2) e^{-\lambda p \pi R_0^2} (1 - p) \\ &= p(1 - p)(1 - e^{-\lambda \pi R_0^2/2}) e^{-\lambda p \pi R_0^2} , \end{aligned} \quad (15)$$

which is in accordance with the results of [5].

2.4 Expected Forward Progress

The forward progress, z , of a successful packet is the distance covered over a single-hop in the direction of the final destination. It has been proven in [5] that:

$$E[z] = \int_0^{R_0} \frac{2rd(r)}{\pi} dr , \quad (16)$$

where $d(r)$ is the pdf of the distance d between a sender and a receiver for a successful transmission (event that we denote (R)). Let us evaluate $d(r)$ in the case of multi-packet reception:

$$P[r \leq d \leq r + dr | R, T_n, D_k] = \frac{P[R|r \leq d \leq r + dr, T_n, D_k]P[r \leq d \leq r + dr | T_n, D_k]}{P[R|T_n, D_k]} . \quad (17)$$

We now make the realistic assumption that if k packets are decoded among n , the successful senders are the k closest senders to the receiver. Under this assumption, for $n > 1$ and $1 \leq k \leq n$:

$$P[R|T_n, D_k] = \frac{k}{n}, \quad (18)$$

$$P[R|r \leq d \leq r + dr, T_n, D_k] = \sum_{i=0}^{k-1} \binom{n-1}{i} \left(\frac{r^2}{R_0^2} \right)^i \left(1 - \frac{r^2}{R_0^2} \right)^{n-1-i}, \quad (19)$$

$$P[r \leq d \leq r + dr | T_n, D_k] = P[r \leq d \leq r + dr] = \frac{2r}{R_0^2} dr . \quad (20)$$

Eq.18 is the proportion of successful transmissions during the considered slot. Eq.19 is justified by the fact that a transmission is successful at distance r from the receiver iff there are at most $k-1$ senders in the disk of radius r . Moreover, the probability for a sender to be in this disk is r^2/R_0^2 . It is straightforward to verify that the integration of Eq.19 over the disk of radius R_0 results in Eq.18. The last equation is the pdf of the distance between any node in the disk of radius R_0^2 and the receiver. It is possible to un-condition Eq.17 by taking into account the Poisson distribution and the $r_{n,k}$ probabilities:

$$P[r \leq d \leq r + dr | R, T_n] = \sum_{k=0}^n P[r \leq d \leq r + dr | R, T_n, D_k] r_{n,k}, \quad (21)$$

$$\begin{aligned} P[r \leq d \leq r + dr | R] &= \sum_{n=1}^{\infty} \sum_{k=0}^n P[r \leq d \leq r + dr | R, T_n, D_k] r_{n,k} P[T_n] \\ &= \sum_{n=1}^{\infty} \sum_{k=0}^n P[r \leq d \leq r + dr | R, T_n, D_k] r_{n,k} \frac{(\lambda p \pi R_0^2)^n}{n!} e^{-\lambda p \pi R_0^2} . \end{aligned} \quad (22)$$

We can now conclude for $d(r)$:

$$d(r) = \sum_{n=1}^{\infty} \sum_{k=1}^n \sum_{i=0}^{k-1} \binom{n-1}{i} \left(\frac{r^2}{R_0^2} \right)^i \left(1 - \frac{r^2}{R_0^2} \right)^{n-1-i} \frac{2nr}{kR_0^2} \frac{(\lambda p \pi R_0^2)^n}{n!} e^{-\lambda p \pi R_0^2} r_{n,k} . \quad (23)$$

The numerical evaluation of Eq.16 implies the following integration:

$$\int_0^{R_0} r^{2(i+1)} \left(1 - \frac{r^2}{R_0^2} \right)^{n-1-i} dr = \sum_{j=0}^{n-1-i} \binom{n-1-i}{j} \frac{(-1)^j}{R_0^{2j}} \int_0^{R_0} r^{2(i+j+1)} dr$$

$$= \sum_{j=0}^{n-1-i} \binom{n-1-i}{j} \frac{(-1)^j R_0^{2i+3}}{1+2(i+j+1)}. \quad (24)$$

As a consequence:

$$E[z] = \sum_{n=1}^{\infty} \sum_{k=1}^n \sum_{i=0}^{k-1} \sum_{j=0}^{n-1-i} \binom{n-1}{i} \binom{n-1-i}{j} \frac{4(-1)^j R_0 r_{n,k} e^{-\lambda p \pi R_0^2} (\lambda p \pi R_0^2)^n}{k \pi (3+2i+2j)(n-1)!} \quad (25)$$

With the help of a software of formal computations, we can simplify this expression in:

$$E[z] = \frac{4R_0 e^{-\lambda p \pi R_0^2}}{\pi} \sum_{n=1}^{\infty} \sum_{k=1}^n \frac{\Gamma(k + \frac{3}{2}) (\lambda p \pi R_0^2)^n r_{n,k}}{3\Gamma(n + \frac{3}{2})k!}, \quad (26)$$

where Γ is the gamma function.

2.5 End-to-End Throughput

According to [5], for any randomly selected terminal, the expected path length between it and another selected terminal is given as $D = (128/45\pi)\sqrt{N/\lambda\pi}$, where N is the number of nodes in the network. Thus, the mean number of hops for a packet is $D/E[z]$ and the end-to-end throughput of the network per slot is:

$$t = \frac{SE[z]}{D}. \quad (27)$$

3 Multi-packet Reception Models

In this section, we assume that the previously considered ALOHA protocol is a spread slotted ALOHA. At a given time-slot, all senders are supposed to choose at random a pseudo-noise (PN) code among a large book of low cross-correlated PN codes with spreading factor L , large. All potential receivers, i.e., all nodes have the knowledge of this book and are able to perform multi-packet reception. We neglect the probability that two neighboring senders choose the same code in order to simplify the calculations. From the presented models, we derive values for the $r_{n,k}$.

3.1 Simple Model

The first model is a very simple one, often used in the literature, e.g., in [9]. It states that all of the simultaneous transmissions can be successfully received if no more than K users are transmitting at the same time. If there are more than K users transmitting at the same time, the multi-user receiver is overwhelmed and a collision occurs. Thus:

$$r_{n,k} = \begin{cases} 1, & \text{if } k = n \text{ and } n \leq K \\ 1, & \text{if } k = 0 \text{ and } n > K \\ 0, & \text{otherwise} \end{cases} \quad (28)$$

In the following two models, a packet is assumed to be decoded by an idle node if its signal to interference plus noise ratio (SINR) reaches a SINR target at the output of the detector.

3.2 Receiver with a Bank of Matched Filters

In this section, we suppose that radio receivers devices are made of a bank of MF that are able to decode each spreading code individually. If P_0 is the transmit power, the received power at a distance r is assumed to be $P(r) = P_0/r^\gamma$, where $\gamma > 2$ is the path loss exponent. This expression is a far-field approximation that doesn't hold for small values of r . A packet is considered to be decoded if the SINR, β , of a signal at the output of the MF reaches a SINR target β_0 , i.e., if:

$$\beta = \frac{P(r)}{\sigma^2 + \frac{1}{L} \sum_{i=0}^{n-1} \frac{P_0}{r_i^\gamma}} \geq \beta_0, \quad (29)$$

where σ^2 is the power of the noise, n is the number of interferers, and L is the spreading length.

In order to analytically evaluate the $r_{n,k}$ parameters, the cumulative distribution function (cdf) of the SINR is needed in the case of a Poisson field of interferers. This problem has been treated in [6] and in [11], where the characteristic function of the interference $Y = \sum_{i=0}^{n-1} P_0/r_i^\gamma$ has been obtained:

$$\phi_Y(\omega) = \exp\left(-\pi\lambda p\Gamma(1-2/\gamma)e^{-i\pi/\gamma}\omega^{2/\gamma}\right), \quad \omega \geq 0 \text{ and } \gamma > 2, \quad (30)$$

where Γ is the gamma function and p is the probability of transmission. This expression leads to the exact cdf of β and thus to the $r_{n,k}$ in the MF case. However, we will see that this is not the case for the MUD receiver. That is the reason why we evaluate the $r_{n,k}$ probabilities thanks to Monte Carlo simulations in order to allow a fair comparison with the MMSE detector.

A Poisson field of interferers with density λp is generated on a two dimensional squared network $[-Xmax; Xmax] \times [-Ymax; Ymax]$. The considered receiver, a , is placed at $(0; 0)$. R_0 is fixed as the maximum distance from which can come packets for the receiver. In the absence of interferer, R_0 verifies the following expression: $\beta_0 = P_0/(R_0^\gamma \sigma^2)$. n is the number of senders inside the disk of radius R_0 with center a . For each of these senders, the SINR is computed after summing the interference from the whole network. If the SINR reaches the SINR target, the packet from this sender is assumed to be decoded. A snapshot of the simulation is shown on Fig.1. Tab.1 shows the parameter values used for our simulations.

Fig.2 shows the plot of the matrix $r_{n,k}$ for $n \leq 14$ and $p = 0.2$. The mean number of senders in the disk of radius R_0 is $\lambda p \pi R_0^2 \approx 5$, so the probability that $n > 14$ is very low. This figure shows that for small values of n , all packets are decoded. Then, when n increases, the number of decoded packets decreases.

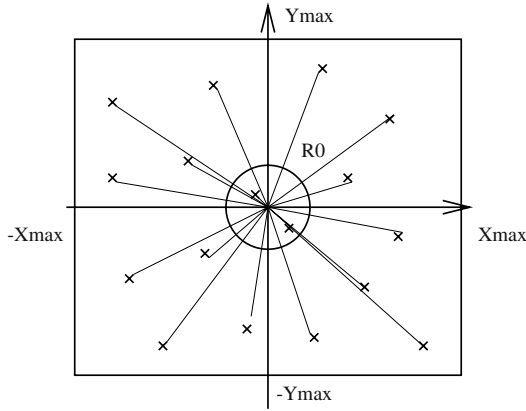


Fig. 1. Snapshot of the Monte Carlo simulation: the power of all the interferers are summed at the receiver.

Table 1. Parameter values used for the Monte Carlo simulation in the case of MF receivers

Parameter	Value
$Xmax$	50
$Ymax$	50
λ	0.25
p	0.2
L	32
P_0	5
β_0	0.025
σ^2	0.2
γ	4

3.3 Receiver with MMSE Multi-user Detector

In this section, we assume that receivers are able to perform multi-user detection thanks to a MMSE detector. While the traditional MF or Rake receiver treats interference from other users as noise, the MUD scheme jointly decodes all users.

The condition of decoding of a packet is still based on the SINR at the output of the signal detector. According to [15], to check if the target for a given sender's SINR, β_0 , can be met for a given system of senders, it suffices to check the following condition:

$$\frac{P}{\sigma^2 + \frac{1}{L} \sum_{i=0}^{n-1} I(P_i, P, \beta_0)} \geq \beta_0, \quad (31)$$

where $P = P_0/r^\gamma$ is the received power of the given sender, P_i is the received power from the interferer i and $I(P_i, P, \beta_0)$ is the effective interference of sender

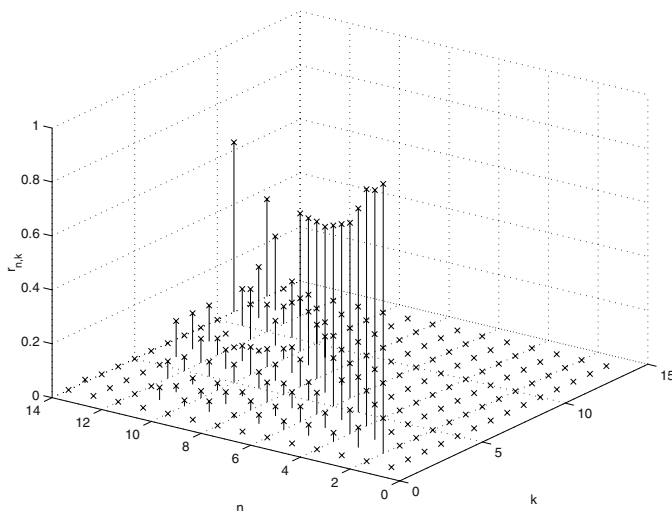


Fig. 2. Probabilities, $r_{n,k}$, for a receiver to decode k packets given that n have been sent in the case of a bank of MF.

i on the considered sender at the target SINR β_0 :

$$I(P_i, P, \beta_0) = \frac{PP_i}{P + P_i\beta_0}. \quad (32)$$

Eq.31, also used in [8] in the context of call admission control, is an approximation since it is true for large systems, when $L \rightarrow \infty$, $n \rightarrow \infty$ and $L/n = \alpha$, and for random spreading sequences.

We can show that the characteristic function of the interference for a given sender and a given SINR target, β_0 is:

$$\phi_Y(\omega) = \exp \left(i\lambda p \pi \omega \int_0^{P/\beta_0} \left(\frac{P_0}{t} - \frac{P_0\beta_0}{P} \right)^{2/\gamma} e^{i\omega t} dt \right). \quad (33)$$

While Eq.30 is seen as the characteristic function of a stable law, Eq.33 seems to be un-tractable for further computations. That is the reason why we rely on Monte Carlo simulations as explained in the previous section. Parameter values are given in Tab.1 and the condition of packet decoding is given by Eq.31. Fig.3 shows the graph of the matrix $r_{n,k}$ for $n \leq 14$. It is clear that the MUD scheme offers much better performances than the MF decoding. Note also that the simple model is a approximation of the MMSE performance if K is chosen appropriately.

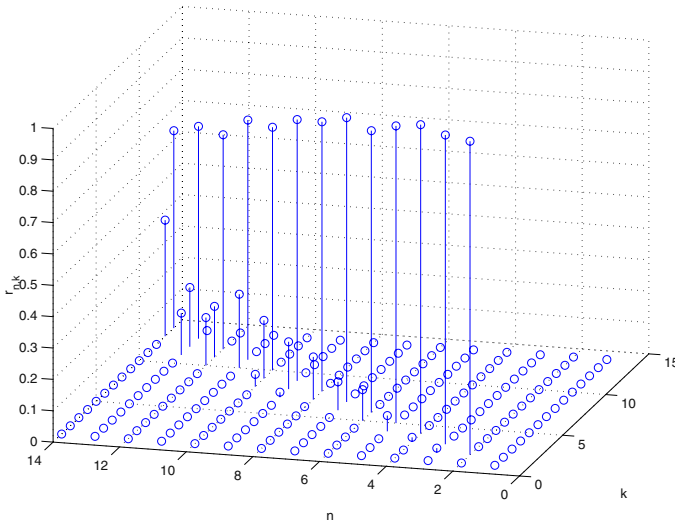


Fig. 3. Probabilities, $r_{n,k}$, for a receiver to decode k packets given that n have been sent in the case of MMSE MUD.

4 Numerical Results

In this section, we give numerical results for the three models of multi-packet reception presented previously. We focus our attention on the local throughput and on the end-to-end throughput of the network.

On Fig.4, we present the local throughput for the first simple model with different values of K . We observe in all cases the characteristic shape of the throughput of the ALOHA protocol as a function of the input load. As expected, the multi-packet reception feature improves the maximum achievable throughput.

Fig.5 shows the end-to-end throughput for the first simple model with different values of K . Here also, we see the advantage of multi-packet reception. Note that the optimum probability of transmission depends on K . For $K = 1$, we observe the classical result that p is optimum for $p = 1/(\lambda\pi R_0^2)$, which here is approximately 0.05. As K increases, p also increases because more packets can be handled by the receiver.

Fig.6 compares the local throughput of the MF receiver with this of the MMSE receiver. We observe the great advantage of the MUD over the conventional receiver (approximately 30% in our scenario). This advantage can also be seen on Fig.7, that shows the end-to-end throughput. Indeed, the joint detection of all users makes the MUD very robust to near-far problems. This near-far resistance is of great interest in ad hoc networks because power control schemes are difficult to implement in such decentralized networks.

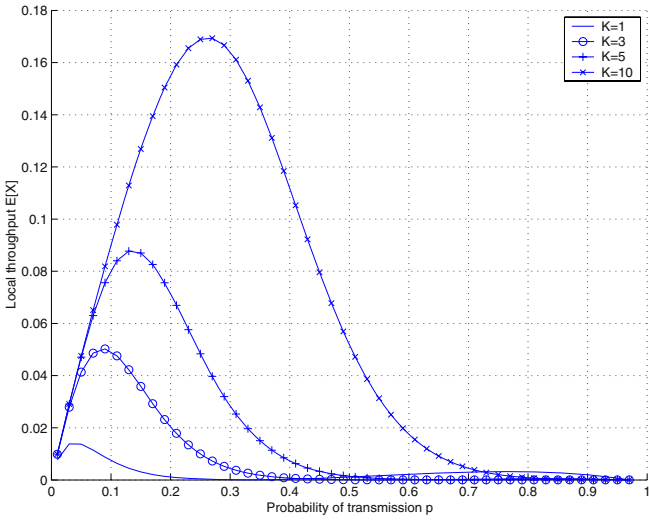


Fig. 4. Local throughput in packets/time-slot for the simple model of multi-packet reception for different values of K , the maximum number of packets that can be decoded by the receiver.

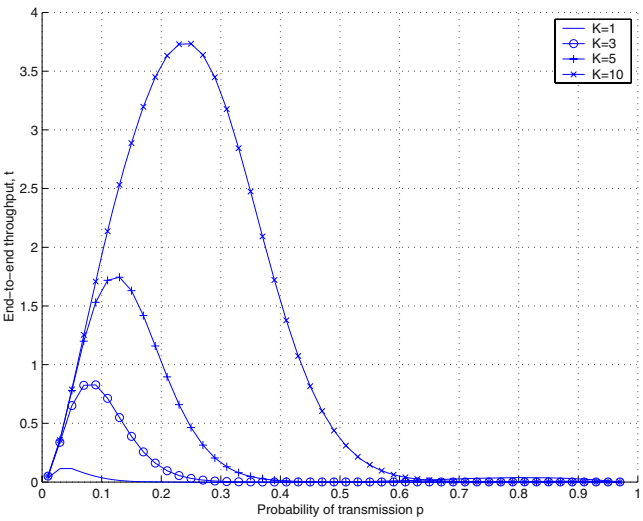


Fig. 5. End-to-end throughput in packets/time-slot for the simple model of multi-packet reception for different values of K , the maximum number of packets that can be decoded by the receiver, and $N = 100$ nodes.

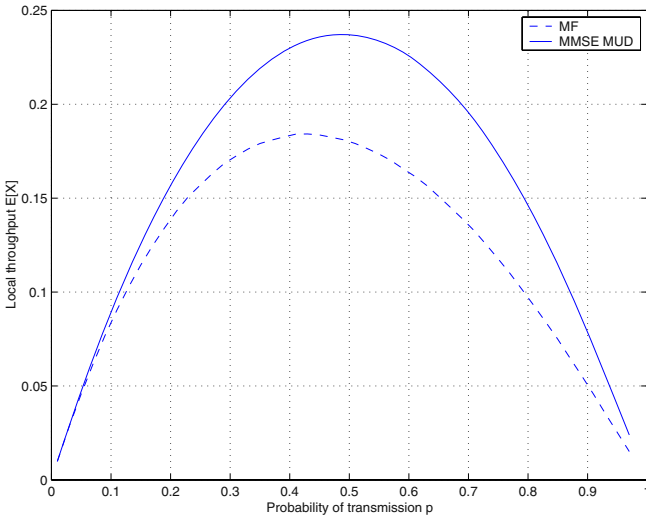


Fig. 6. Local throughput in packets/time-slot for the MF receiver and the MMSE receiver.

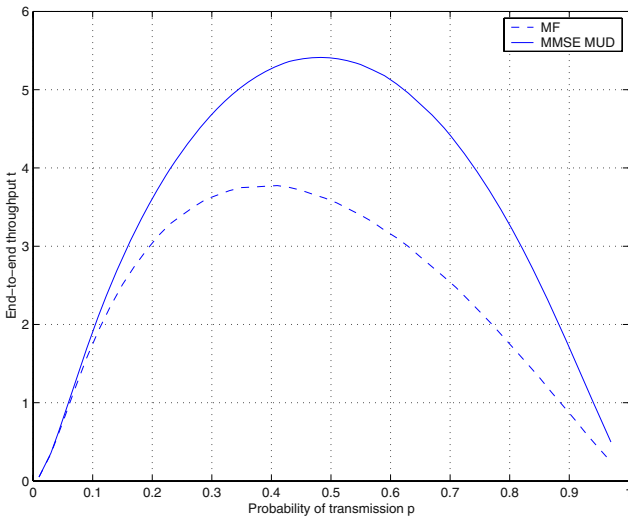


Fig. 7. End-to-end throughput in packets/time-slot for the MF receiver and the MMSE receiver for $N = 100$ nodes.

5 Conclusion

In this paper, we have analyzed the throughput of the slotted ALOHA protocol in a multi-hop network, where nodes are able to perform multi-packet reception. We have derived from an analytical study a closed-form formula for the local throughput and the end-to-end throughput of such a network. This formula is given as a function of the probabilities, $r_{n,k}$, of decoding k packet when n senders have transmitted a packet in the neighborhood of the receiver. Then, three models of multi-packet reception have been presented in the case of CDMA systems: a simple one, often used in the literature, and two models based on two types of receivers, i.e., a bank of matched filters and a MMSE MUD detector. In the latter case, we have provided the characteristic function of the interference. However because of the un-tractability of this formula, we relied on Monte Carlo simulations in order to evaluate the $r_{n,k}$ probabilities. Numerical results show the great advantage of the near-far resistance of the MMSE receiver.

References

1. P. Karn, MACA - a New Channel Access Method for Packet Radio, Proc. of ARRL/CRRL, April 1990.
2. V. Bhargavan, A. Demers, S. Shenker, and I. Zhang, MACAW: A Media Access Protocol for Wireless LAN's, Proc. of ACM SIGCOMM, pp 212-225, Aug. 1994.
3. J. J. Garcia-Luna-Aceves and C. L. Fullmer Floor Acquisition Multiple Access (FAMA) in Single-Channel Wireless Networks Mobile Networks Applications, vol. 4, pp 157-174, Baltzer Science Publishers, 1999
4. IEEE P802.11, Draft Standard for Wireless LAN: Medium Access Control (MAC) and Physical Layer (PHY) Specifications IEEE, July 1996.
5. R. Nelson and L. Kleinrock The Spatial Capacity of a Slotted ALOHA Multihop Packet Radio Network with Capture, IEEE Trans. On Communications, VOL. COM-32, NO. 6, June 1984
6. E. S. Sousa and J. A. Silvester, Optimum Transmission Ranges in a Direct-Sequence Spread-Spectrum Multihop Packet Radio Network, IEEE Journal On Selected Areas in Communications, VOL. 8, NO. 5, June 1990
7. E. S. Sousa, Performance of a Spread Spectrum Packet Radio Network Link in a Poisson Field of Interferers, IEEE Trans. On Information Theory, VOL. 38, NO. 6, Nov. 1992
8. C. Sankaran and A. Ephremides, The Use of Multiuser Detectors for Multicasting in Wireless Ad Hoc CDMA Networks, IEEE Trans. On Information Theory, VOL. 48, NO. 11, Nov. 2002
9. Q. Liu, E.-H. Yang, and Z. Zhang, Throughput Analysis of CDMA Systems Using Multiuser Receivers, IEEE Trans. On Communications, VOL. 49, NO. 7, July 2001
10. A. Polydros and J. Silvester, Slotted Random Access Spread-Spectrum Networks: An Analytical Framework, IEEE Journal On Selected Areas in Communications, VOL. 5, NO. 6, July 1987
11. E. S. Sousa, Interference Modeling in a Direct-Sequence Spread-Spectrum Packet Radio Network, IEEE Trans. On Communications, VOL. 38, NO. 9, Sept. 1990
12. J. Q. Bao and L. Tong, A Performance Comparison Between Ad Hoc and Centrally Controlled CDMA Wireless LANs, IEEE Trans. On Wireless Communications, VOL. 1, NO. 4, Oct. 2002

13. R. Lupas and S. Verd, Near-Far Resistance of Multiuser Detectors in Asynchronous Channels, IEEE Trans. On Communications, VOL. 38, NO. 4, Apr. 1990
14. S. Verdu, Multiuser Detection Cambridge University Press, 1998
15. D. N. C. Tse and S. V. Hanly, Linear Multiuser Receivers: Effective Interference, Effective Bandwidth and User Capacity, IEEE Trans. On Information Theory, VOL. 45, NO. 2, Mar. 1999
16. A. Frey and V. Schmidt, Marked Point Processes in the Plane I, Advances in Performance Analysis, VOL. 1, Notable Publications Inc., 1998

WIDE: Wireless Information Delivery Environment in Distributed Hot Spots*

Mehmet Yunus Donmez, Sinan Isik, and Cem Ersoy

Bogazici University, Department of Computer Engineering,
Istanbul, Turkiye
{donmezme, isiks, ersoy}@boun.edu.tr

Abstract. We developed an information delivery system, namely WIDE (Wireless Information Delivery Environment), in client-server architecture using 802.11b infrastructure. WIDE aims to deliver popular information services to registered mobile clients in WLAN hot spots. We present the proposed system architecture, related delivery mechanism and communication protocols. We also give a brief overview of mechanisms required for secure and reliable communication in WIDE system. Performance evaluation results of the proposed system using the implemented prototype are also included in this paper.

1 Introduction

Current advances in computer technology lead to the emergence of battery-operated, low-cost and portable computers such as personal digital assistants (PDAs) or laptop computers equipped with wireless communication peripherals. The increasing demand to access data stored at information servers even while the users are on the move, coupled with the continuing advances in telecommunications, interconnectivity and mobile computing, make information delivery to mobile clients a broadly studied subject.

The motivation of the system proposed in this paper is the Infostation concept [1]. An Infostation is seen as a small cell providing a high bandwidth radio link for data services, which takes on the concept of discontinuous service provision for certain types of services. There are ongoing projects on Infostation concept. One of these project, which is being carried on by Rutgers University, focuses primarily on the data link layer and below [2]. Another project, which is being carried on by Polytechnic University, focuses on developing Infostation applications and transport layer mechanisms to support those applications [3,4]. Rover Technology [5] is another project, which is studied at MIND Laboratory in University of Maryland employing location tracking to decide the services to deliver to the users.

In this paper, we describe the design of a system, namely WIDE (Wireless Information Delivery Environment), which delivers popular or personal information

* This work is partially supported by the State Planning Organization of Turkey under the grant number 98K120890, and by the Bogazici University Research Projects under the grant number 02A105D.

services to registered mobile clients in wireless hot spots. The system design includes protocols that use the IEEE 802.11b WLAN technology to distribute data within isolated coverage areas in a reliable and secure manner.

WIDE resembles gas stations or ATM machines [4], which can be found in locations where there is the appropriate user density, but where users drive or walk to in order to actually access the service. As users pass through the coverage area of the system the most recent version of the subscribed information services will be automatically downloaded to their mobile terminals without any user intervention. Received data may be used in a later time.

In a campus environment, as the user passes through the hot spot of a department building with his PDA or laptop computer, the system may be used to download a wide range of information that might be useful to him. This information may include the most recent data about course locations, course announcements, course web pages and course notes as well as the events in the building and on the campus. As the user walks out of the building and arrives to the cafe, information relevant to that environment such as administrative, departmental, student club and cultural organization announcements are delivered to the user as well as newspaper articles, e-books, etc.

2 WIDE System

2.1 Requirements of WIDE

WIDE is a client-server system, which aims to deliver wireless information services to registered mobile and stationary clients in distributed hot spots. The design of the WIDE system has to meet some basic requirements. First of all, clients of WIDE must be authenticated by the system before getting any services. For this purpose, a secure authentication mechanism should be employed in WIDE. In addition, a global security mechanism should be employed on WIDE, so that the network packets of WIDE system should only be identified and processed within WIDE components.

The transfer of any information services has to be arranged in a way that requires little or no human-computer interaction while users pass through the coverage area of a server. Also, transmissions to and from clients should be arranged so that the desired transactions can be completed while clients are in the coverage area. In addition, any updates of the information services offered to the client shall be transmitted to clients. Hence, we need to have information about the interests of the authenticated clients. A publish/subscribe mechanism should be designed to create a user profile for each client of WIDE system. Subscriptions of clients to information services are recorded in their user profiles. Clients receive information services in case of any updates with the help of their user profiles as they pass through the wireless coverage area of a server.

Since the system offers popular information services, it should perform in an acceptable level in terms of reception time for individual clients when there are many users demanding the same service. Hence, the design should be based on data broadcasting, or, more precisely data multicasting to provide scalability and efficient use of the wireless channel.

The protocols included in the system have to be designed with the idea of battery energy conservation. In addition, they must satisfy the reliability needs of the wireless medium. The residence time of a client in the coverage area may be very short which may lead to an incomplete data transfer. The completion of any incomplete data transfer should be dealt by the system infrastructure using some recovery and error correction mechanisms. In addition, the protocols for data transfer should be designed in a way that allows the coexistence of other communication traffic on the wireless channel.

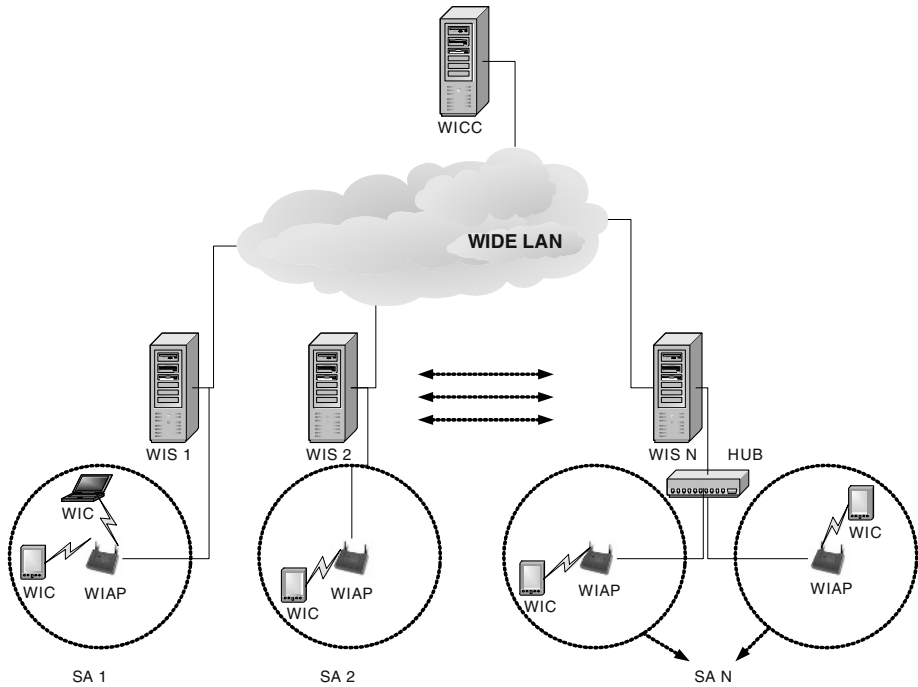


Fig. 1. WIDE system architecture

2.2 WIDE System Architecture

There are three main components in WIDE system as shown in Figure 1. These components are clients, data delivery servers and a server controller. We call a client of the system as WIDE Client (WIC), which is a battery operated handheld or laptop PC with necessary equipments that provide wireless connectivity to servers of the system via 802.11b WAPs, namely WIDE Access Points (WIAPs). The servers of the system are called WIDE Server (WIS) and these are responsible for preparing and delivering information services to clients. The information services, which are available for delivery to clients, are assumed to be stored on local disk of each WIS. The delivery management information such as service identifier, class, version, name

and location on the local disk is recorded in a database called WIDE Server Database (SDB).

The component called WIDE Cluster Controller (WICC) keeps and manages system management database called WIDE Cluster Controller Database (CCDB), which consists of a number of tables. These tables are user authentication table, servers information table, user profiles table and information services table.

In WIDE system, each WIS communicates with WICC through the WIDE LAN. The communication between a WIS and a WIC is established via a WIAP. There can be one or more WIAPs connected to a WIS, but a WIAP can only be connected to one WIS. We define the Service Area (SA) of a WIS as the geographical area covered by WIAPs that are connected to a WIS. Figure 1 shows the system architecture, its components and the WIDE LAN.

2.3 Communication Protocols in WIDE

Network Layer Protocols. The system is designed on top of Internet Protocol (IP) stack. WICs must have an IP address that is valid in the SA of a WIS to communicate with that WIS. WIDE is designed to operate independent from how the IP level connectivity of WICs with WISs is established.

Since WICs of WIDE are mobile, they may roam into SA of different WISs, which are probably in different subnets. The IP address assigned to a WIC in one SA may not be valid in another one due to network addressing. DHCP [6] may be used as a valid addressing of WICs in SAs. When WICs enter the SA of a WIS, a DHCP server assigns an IP address to that WIC and the configuration of address is done automatically on the WIC. Here, WIS may be configured as a DHCP server or another machine can be employed as a DHCP server.

Since the residence time of WICs in SAs may be short, a rapid address configuration, such as DRCP [7], may be employed. If IPv6 [8] is chosen as a network layer protocol, stateless autoconfiguration solves the addressing problem of WICs. If Mobile IP [9] is employed in WIDE, each WIS may be configured as a foreign agent or another machine can be employed as a foreign agent. The care-of address assigned by foreign agent will enable WICs to receive service from the system.

Transport Layer Protocols. In WIDE system, WIC-WIS communication is constructed on top of UDP. We cannot deliver popular data to multiple users simultaneously using TCP since it does not support broadcasting and multicasting.

We used IP unicast, IP broadcast and IP multicast mechanisms which are implemented over UDP. IP unicast is required for control messages concerning only the WICs that they are sent to or initiated from. IP broadcast mechanism is employed on the WIS to send control messages that concerns all WICs in the SA. IP multicast mechanism is employed to transfer data simultaneously to multiple users, which are interested in that information.

For the WIS-WICC communication, TCP protocol is used. Between these two components, control messages concerning the administrative databases of the system are transferred. However, these messages are crucial for the system integrity and we have to make sure that they reach to the recipient.

2.4 WIDE Communication Design

Communication between a WIS and WICs proceeds in cycles called Communication Cycles (CCs). In each CC, there are specific time periods in which certain tasks are performed. These time periods, which are named as index broadcast period (IBP), reception preparation period (RPP), data period (DP), authentication period (AUP) and request period (RQP), sequentially follow each other in this order in time. DP is also divided into time slots, which are called as communication slots (CS). Figure 2 illustrates the timing diagram of a CC.

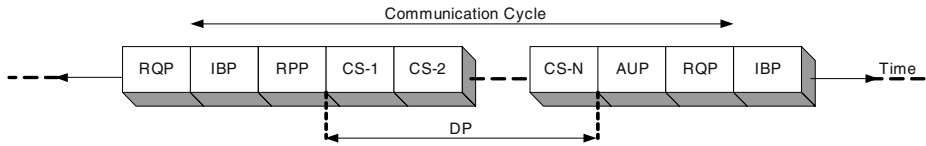


Fig. 2. Timing diagram of a CC

A WIC entered to the SA of a WIS, sends its authentication request to WIS in AUP to be able to receive service from the system. WIS sends the response to authentication request in AUP of a following CC.

Requests of WIC for subscription to information services or requests for unsubscriptions are transmitted to WIS on RQP. In addition, retransmission request for the information services, whose packets are missed, and polling request for updates of information services on the user profile are also transmitted to WIS on RQP. WIS sends the corresponding response messages to WIC on the same RQP.

A scheduler running in WIS decides data to transmit during each CC and prepares the index. The scheduling of an information service in a WIS, requires at least one WIC in the SA of that WIS who previously subscribed or has just subscribed to that service. If WIC has just subscribed to that information service or a retransmission is requested for that service from WIS due to incomplete reception, then that service is queued for delivery. In addition, if WIC has made an authentication or polling request and if there exists a version of that information service that is newer than the one recorded in the user profile of that WIC, then that service is queued for delivery. At the time of delivery, the service appears on the index.

When a WIC is within the SA of that WIS, it listens to the index sent on IBP to see which information services are offered by WIS during that CC. This index message also informs the clients interested with the information service about the multicast group of transmission and the version of the data to be transmitted. Each multicast group is coupled with a CS in DP. Application program running on WIC examines the index and determines whether there are any available items of interest by examining the user profile existing in the mobile terminal. If items of interest are available, WIC performs the necessary operations such as joining to the announced multicast group and preparing the buffers for receiving an information service in the RPP. Information services are delivered to WICs in the form of packets of fixed size. Data packets of each item announced for that CC in the index are delivered in the corresponding CS in DP. Consequently, WIC will receive data packets of the

interested service from the joined multicast group, while other data packets will be dropped by the IP layer.

Data communication between WICs and WISs is established by using IP broadcast, IP multicast and IP unicast technologies together. There are several virtual channels utilizing the above technologies. These channels do not physically exist and they share the same physical channel of 802.11b. Virtual channels are defined in the transport layer and can be classified as point-to-point channels, broadcast channel and multicast channels as identified by their corresponding IP addresses and port numbers.

There is only one Broadcast Channel (BCH). Point-to-point channels used in the system are named as Uplink Authentication Channel (UACH), Downlink Authentication Channel (DACH), Uplink Request Channel (URCH) and Downlink Request Channel (DRCH). Data Channels (DCHs), whose number is a predefined system parameter, are multicast channels and each DCH has its own communication slot. The virtual channels, time periods and the messages between WIC and WIS are illustrated in Figure 3.

BCH can be considered as a control channel. WIS sends the index message on the period IBP; start and finish probes announcing the beginning and end of AUP and RQP periods. When a WIC enters to the service area of a WIS, it is informed about the existence of that WIS and then synchronizes with the communication cycle of that WIS by listening to BCH.

UACH and DACH are used for the authentication of WICs. WICs send their authentication requests over UACH on the periods AUP and WISs send authentication notification messages to WICs over DACH on the AUP periods.

URCH and DRCH are used for subscription, unsubscription, retransmission requests and notification purposes. WICs send subscription to an information service request, unsubscription from an information service request, periodic polling request, retransmission of an incomplete transfer request, over URCH on the periods RQP. WISs send notification of these requests in the same RQP.

The time period of each DCH corresponds to a CS, which appear in a DP one after the other. DCHs are used to deliver information services to WICs on the corresponding CS in period DP. Index message informs WICs about the DCH over which the information services will be delivered in that CC. WICs join the IP multicast groups in the period RPP, if the information services announced are the interested ones. They start to receive UDP packets including the bytes of the interested information service on the corresponding period in DP in that CC.

WIS and WICC communicate with each other over a full duplex TCP channel. The communication between these two components of the WIDE system is realized by messages exchanged over this channel.

2.5 Mechanisms of WIDE System

Publish / Subscribe Mechanism. Subscription to information services is provided with a publish/subscribe mechanism in WIDE system. The list of the information services offered by the system is called the table of contents (TOC), which is also offered as a service. In WICs, a user interface is provided to view the local copy of

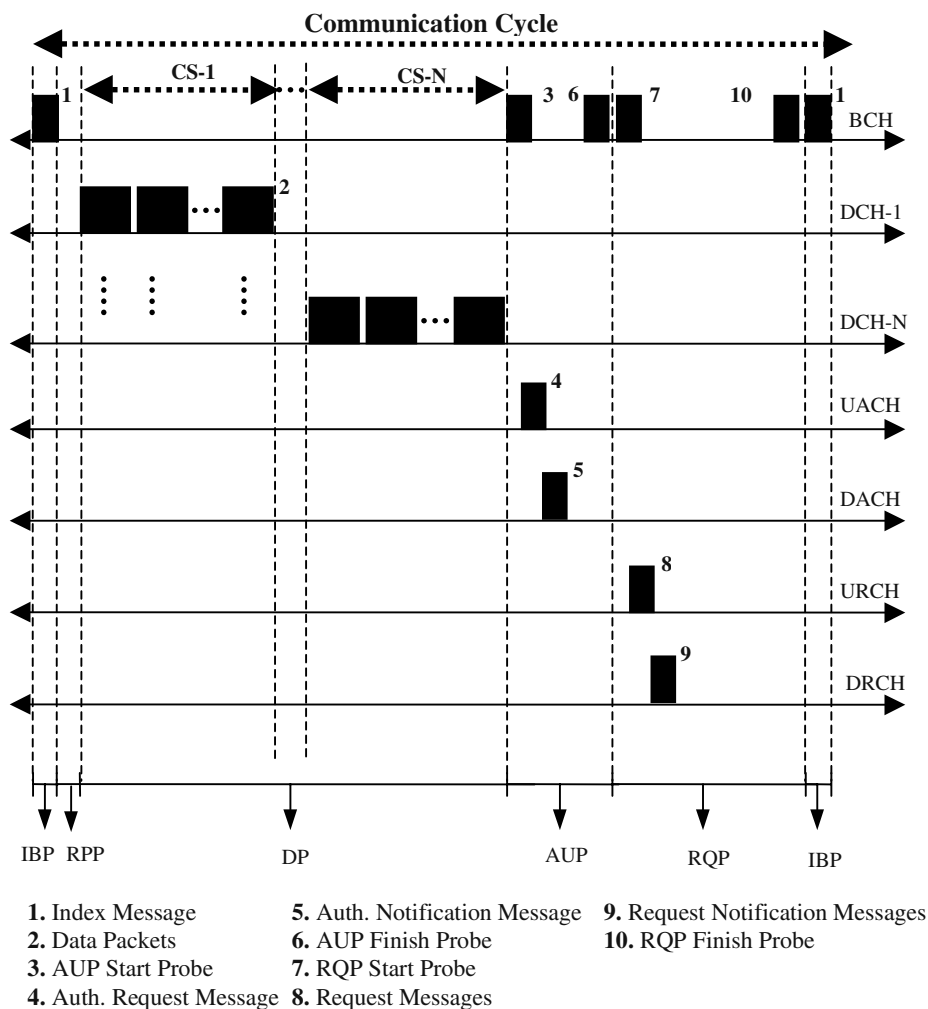


Fig. 3. Communication cycle and virtual channels

TOC to users. Figure 4 shows the TOC GUI on a PDA client. Users may create a subscription request anytime and anywhere with the help of this user interface. Subscription requests are transmitted automatically to WICC via a WIS when WIC roam into the SA of that WIS. A local and a remote list of subscriptions are kept in WIC and WICC respectively. Initially the TOC service is automatically in the subscription list for each WIC. Remote subscription list is kept as a user profile for each WIC.



Fig. 4. Local copy of Table of Contents on a PDA client

Similarly, if user does not want to receive or update a service any more, he can create an unsubscription request with the help of TOC user interface. Unsubscription requests are also transmitted automatically to WICC via a WIS when WIC roam into the SA of that WIS. The entry for the service that the user wants to unsubscribe from is deleted from the corresponding user profile.

Reliable Data Delivery Mechanism. The messages initiated by the WIC have to be acknowledged by the WIS. These messages are the request messages related to information services. WIC has to be sure that its requests are received by the WIS and they are being processed in the system. The functionality of the system depends on the reliable delivery of these messages to WIS. If acknowledgement messages for information service related requests are not received by the WIC in the same request period, then it repeats its requests in the next request period. Similarly, if acknowledgement messages for authentication requests are not received by the WIC in a predefined duration, then it repeats its requests in the next authentication period after the expiration of the duration.

We choose to employ a reliability mechanism, which employs a mixed type of carousel [10], erasure code and automatic retransmission request (ARQ) [11] techniques. We do not require the reliable transmission of each data packet individually. Before transmission of data packets of an information service, these packets are encoded using a forward error correction (FEC) technique called erasure codes in which the reception of any k packets out of $k+m$ transmitted packets is sufficient for reliable reception [12]. After this phase, a packet number is given to each packet in sequential order. WIC keeps track of the packets that it received using these packet numbers. This helps WIC to discard the duplicate packets caused by the carousel mechanism. When enough number of packets is received for the FEC

decoding process, these packets are decoded to form the actual data packets in sequential order. If the received number of packets is not enough to recover the actual data, the missing packets can be captured in the next carousel cycle, if exists. If there are still missing packets to be captured, then an ARQ request is prepared by WIC to request the retransmission of that information service.

Security Mechanism. Security mechanisms include confidentiality, entity authentication, data origin authentication and data integrity. Symmetric-key encryption schemes are used to provide security in WIDE system. These encryption schemes have low complexity and high data throughput, providing fast and power-efficient processing [13]. The contents of each packet exchanged between WISs and WICs has to be kept secret from the public. Hence, each packet has to be encrypted with a key, which is only known by the endpoints of the communication and WICC. Each component in the system has a distinct key. WICC key is known by all of the WICs in the system. The headers of each packet initiated from a WIS are encrypted with the WICC key to be identified by each WIC including unauthenticated ones in the SA. The payload parts of these packets are encrypted with the WIS key which serves as a service key. The WIS key is acquired by WICs at the end of entity authentication operations. Entity authentication is accomplished in WICC by comparing the user password ciphered with WIC key in authentication request message with the correspondent in the user authentication table.

Authentication also applies to information itself. WIS and WICs entering into a communication should identify each other. Information delivered over a channel should be authenticated as to origin, date of origin, data content, time sent, etc. We used time stamping for each packet passed between WICs and WISs. Each party in communication checks the time stamps of the messages, which are received from other parties, and keeps the last received time stamp for each different party. The packets, which have time stamps earlier than or equal to the last encountered time stamp, are discarded. This prevents the situations in which any fake server captures the packets and delivers these copies or modified versions of these packets. Additionally, a two way challenge-response mechanism is applied to all requests and responses. WIS announces a challenge for AUP and RQP in start probes. WIC puts the response of that challenge together with its own challenge in the request message. In the notification message, WIS sends the response of challenge in the request message back to WIC. This mechanism ensures that the responding party is the one that is expected to respond. A challenge is a random number generated for each CC. There are distinct challenge functions in both WIC and WIS known by each other. If the response sent to the initiator of a challenge is the same as the result of the challenge function of the responding party, then that packet is recognized as a WIDE packet.

3 Implementation of a WIDE Prototype and Its Performance

We implemented a WIDE prototype, which delivers services in a campus environment. Initial prototype uses the Bogazici University, Computer Engineering (CMPE) WLAN access points as WIAPs. The components are implemented using

Microsoft Visual C++ 6.0 and Microsoft Platform Software Development Kit (SDK) for Visual C++ 6.0. WIC prototype is designed to run on laptop computers, which have Windows 98 Second Edition or later operating system on them for full functionality. WIS and WICC prototypes run on desktop machines which have a Windows 2000 Family operating system on them. In addition, we have a WIC prototype having partial functionality that runs on a Toshiba E740 PDA with Pocket PC operating system.

We designed some experiments to evaluate the performance of WIDE prototype. In these experiments, we executed the WIC application on a laptop computer with a Pentium III processor operating at 500 MHz and 192 MB RAM. The operating system on WIC machine is Windows 2000 Professional Edition.

WIS and WICC applications are executed on the same desktop computer with a Pentium IV processor operating at 1800 Mhz and 1 GB RAM. The operating system on server machine is Windows 2000 Advanced Server Edition.

The wireless connectivity between server and the client is provided with a Cisco AiroNet 350 Series WAP connected to the server machine via an Ethernet adapter and a 3Com AirConnect WLAN PCMCIA card installed on the client.

In our experiments, we set the data payload size to a value of 1400 bytes because the maximum throughput of UDP on wireless medium is achieved when the UDP packet size is 1472 bytes and rapidly drops after this value [14]. Each data packet contains the header of its own which is added by the WIS application. The sum of data packet payload, data packet headers and UDP headers should be less than 1472 bytes.

Typically, number of carousel cycles is set to a value of two. However, in our tests, we set this parameter to a value of one to be able to detect the number of cycles for the completion of the reception of an information service.

The length of the time periods in a CC are also kept fixed in the tests. The durations of RPP, AUP and RQP are set to 100 ms, 10 ms and 10 ms respectively on WIS. For these experiments, we prepared files of size varying from 100 KB to 1000 KB with an increment of 100 KB.

3.1 Effect of Socket Buffer Size and DBDP

The client machine is placed indoor, to five meters apart from the WAP, in the line of sight, to evaluate the effect of buffer size and *DBDP* on the performance. *DBDP* is the delay placed between each data packet. The requested information service was a file of size 100 KB. The socket buffer sizes were varied between the Windows default value of 8 KB to 256 KB increasing with powers of two. We measured the time between the reception of the first packet and the last packet of the file on the application layer. The experiments were repeated for different values of *DBDP* varying between 0 ms to 30 ms with an increment of 10 ms. In Figure 5, socket buffer sizes are plotted versus reception times for different *DBDP* values. Each data value on the graph corresponds to the average of 15 experiment results.

Figure 5 presents that the buffer sizes of sockets used for information service reception in client prototype significantly affect the performance. With the default buffer size, even if a delay of 20 ms is placed between data packets, packet losses occur because of buffer overruns. For a delay of 30 ms, read operations from the buffer is faster than write operations to the buffer. In this case, we get a straight line

because there are not any buffer overruns. However, reception of 100 KB in two seconds is a poor performance. We tried to find a value pair of *DBDP* and buffer size such that there are not any buffer overruns but the reception time is kept as low as possible. It can be observed from the figure that the best possible solution in our experiments is to increase the buffer size to 256 KB and keeping the *DBDP* parameter value at 0 ms when the size of information service is 100 KB.

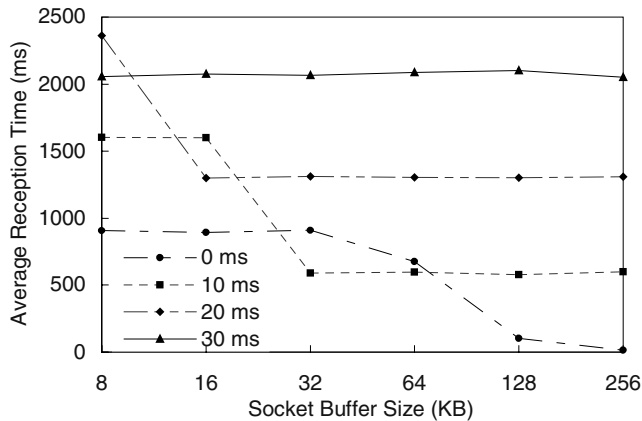


Fig. 5. The effect of buffer size and *DBDP* on average reception time

For higher file sizes, we could not achieve the same performance in terms of reception time with *DBDP* set to 0 ms. In Figure 6, the effect of *DBDP* values on performance in terms of reception time is presented. The setup for this experiment is the same as the previous one. The file size of the information service used in our experiments was 200 KB. Here, we fixed the buffer size of sockets to 256 KB. Each data value on the graph corresponds to the average of 15 experiment results.

If we compare the reception time values for 100 KB and 200 KB when the buffer size is 256 KB and *DBDP* is zero, we observe that there is a large gap between the reception time values. At this point, we analyzed the packets transmitted by IS and packets transmitted by WAP with a network analyzer [15]. We observed that although all the packets of the file were transmitted by the IS, some of the packets were not transmitted by the WAP. We also analyzed the statistics provided by the WAP to see that the existence of buffer overruns. Hence, we concluded that we should decrease the packet arrival rate to WAP by increasing *DBDP*. In this experiment, we observed that the best performance was achieved when *DBDP* value is around 3 ms. In the range between 0 ms to 3 ms decreasing packet loss led to increase in performance. After 3 ms, in a condition when there is no packet losses, increase in *DBDP* value led to extra delay between packets and hence, a decrease in performance.

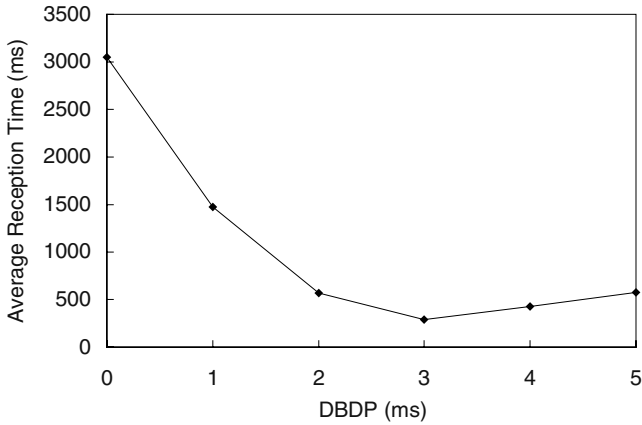


Fig. 6. The effect of DBDP on average reception time for 200 KB file

3.2 Effect of Information Service Size

We repeated experiments for all file sizes between 100 KB and 1000 KB by setting *DBDP* value to 3 ms to see the effect of file size on reception time. Results of these experiments are presented in Figure 7. In these experiments, we measured the reception time at the application layer, which is the elapsed time between reading the first packet and the last packet from the socket buffer, which is represented by application layer data reception (ADR) line. In addition, we measured the elapsed time between the occurrence of the first packet and the last packet on the WLAN with the help of the network analyzer, which is represented by network layer data reception (NDR) line. To be able to compare the achieved performance with another protocol, we downloaded the same files from the FTP server installed on the IS machine. The reception time results given by the FTP client is represented by FTP line. Each data value on the graph corresponds to the average of 15 experiment results.

The results show the expected behaviour for these three cases, which is the linear increase of reception time with the file size. We observe that the packets are not read from the socket buffer as soon as they arrive to the buffer. The most reasonable cause of this behaviour is the context switches to or from the data reception threads forced by the operating system. Because of the latency in reading the packets from the buffers, the packets accumulate in the buffers prior to any read operations. At the application layer, the reading speed is faster than the arrival rate of packets. Hence, the reception time in NDR is observed to be higher than the reception time in ADR. In addition, we observe that NDR performs similar to FTP. The average throughput is calculated as 3.84 Mbps for NDR and 3.83 Mbps for FTP.

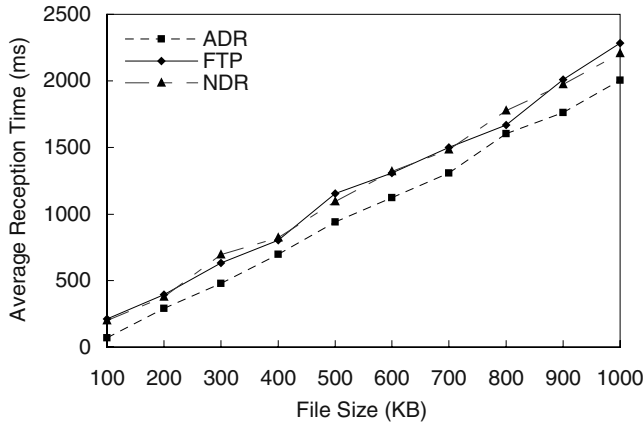


Fig. 7. Average reception time comparison with FTP for different file sizes

4 Conclusions and Future Work

WIDE is a data delivery system, which aims to offer popular information services to mobile clients using a distributed hot spot WLAN infrastructure. The requirements of the system are summarized and the system architecture is introduced. The protocols that WIDE is constructed above are discussed and the details of the communication design between components of WIDE are given. In addition, the mechanisms required for reliable and secure communication and data delivery are presented briefly. We also give some preliminary results of the performance evaluation on the implemented prototype of WIDE system. The initial prototype of WIDE client is implemented for Windows 98 SE or later platforms. The PDA version of WIC prototype should be improved to have full functionality.

Scalability and robustness of the WIDE system are not detailed in the current design. The primary goals of the current implementation were to prove the utilization of the system in a moderate-sized environment such as a university campus and find out the pros and cons of the current architecture. The future goal is to improve the architecture in terms of scalability and robustness. For this purpose, it is being planned to add backup authentication and profiling services to the current design. With these additions, the system will be able to operate under heavy load without affecting the performance dramatically.

We plan to give location based information services to clients. For this purpose we need to find the physical location of the client. We will integrate WIDE with WLAN Tracker [16] and give location based information services. Currently WIDE offers file delivery services. In the future framework, the system can be improved to give streaming and upload services such as music and video streaming, and e-mail transfer requiring special coding and security issues.

References

1. Frenkiel, R.H., Badrinath, B.R., Borras, J., Yates, R.D.: The Infostations Challenge: Balancing Cost and Ubiquity in Delivering Wireless Data. *IEEE Personal Communications*, pp. 66–71, April 2000.
2. DATAMAN Laboratory: NIMBLE: Many-time, Many-where Communication Support for Information Systems in Highly Mobile and Wireless Environments. <http://www.cs.rutgers.edu/dataman/nimble/>, 2003.
3. WICAT, Polytechnic U.: Infostation Project. <http://wicat.poly.edu/infostation.htm>, 2003.
4. Frankl, P., Goodman D.: Technical Overview of the Infostation Project. <http://cis.poly.edu/research/infostation/InfostationOverview.pdf>, 2001.
5. Banerjee, S., Agarwall, S., Kamel, K., Kochut, A., Kommareddy, C., Nadeem, T., Thakkar, P., Trinh, B., Youssef, A., Youssef, M., Larsen, R.L., Shankar, A.U., Agrawala, A.: Rover: Scalable Location Aware Computing. *IEEE Computer*, Vol. 35, No. 10, pp.46–53, 2002.
6. Droms, R.: Dynamic Host Configuration Protocol. IETF RFC 2131, March 1997.
7. McAuley, A., Das, S., Madhani, S., Baba, S., Shobatake, Y.: Dynamic Registration and Configuration Protocol. <http://hnmclab.csie.chu.edu.tw/~tmc/sip/draft-itsumo-drcp-01.txt>, 28 May 2003.
8. Thomson, S., Narten, T.: IPv6 Stateless Address Autoconfiguration. IETF RFC 2462, 1998.
9. Perkins, C.E.: Mobile IP. *IEEE Communications Magazine*, vol. 3, pp. 84–99, May 1997.
10. Acharya, S., Franklin M., Zdonik, S.: Balancing Push and Pull for Data Broadcast. *Proceedings of ACM SIGMOD Int. Conf. on Management of Data*, pp. 183–194, 1997.
11. Rizzo, L., Vicisano, L.: RMDP: an FEC-based Reliable Multicast Protocol for Wireless Environments. *Mobile Computing and Communications Review*, Vol. 2, No. 2, pp. 1–10, April 1998.
12. Rizzo, L.: Effective Erasure Codes for Reliable Computer Communication Protocols. *ACM Computer Communication Review*, Vol. 27, No. 2, pp. 24–36, April 1997.
13. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: *Handbook of Applied Cryptography*. CRC press, 1996.
14. Vasan, A., Shankar, A.U.: An Empirical Characterization of Instantaneous Throughput in 802.11b WLANs. <http://www.cs.umd.edu/~shankar/Papers/802-11b-profile-1.pdf>, 2003.
15. TamoSoft Inc.: CommView 4.0 Evaluation Version. <http://www.tamofiles.com/cv4.zip>.
16. Komar, C.: Location Tracking and LBS in IEEE 802.11 Using WLAN Tracker. *WLAN Tracker Technical Report*, Bogazici University, TR-021, June 2003.

Smart Wireless Access Points for Pervasive Computing

Roel Ocampo^{1,2} and Hermann de Meer^{2,3}

¹ Department of Electrical and Electronics Engineering, University of the Philippines,
Diliman, Quezon City, 1101 Philippines

² Department of Electronic and Electrical Engineering, University College London,
Torrington Place, London WC1E 7JE
{r.ocampo, h.demeer}@ee.ucl.ac.uk

³ Faculty of Mathematics and Computer Science, University of Passau,
94032 Passau, Germany

Abstract. Research in ubiquitous computing has traditionally focused on sensing and making use of user-related context. However, a pervasive computing environment is also a mobile computing environment, where QoS adaptation is often a major concern. We present an integrated framework applicable to both user- and network-oriented adaptation, that uses a common base for sensing context. Smart wireless access points called M-WASPs provide a wireless network infrastructure, and perform sensing, adaptation and other cognitive functions in a distributed fashion, within a pervasive computing environment.

1 Introduction

More and more computing power is being integrated into everyday appliances, which in turn are increasingly being networked. We share the pervasive computing vision described by Mark Weiser in 1991 [1], a future environment where computing power and computing services will be found everywhere, but paradoxically, will increasingly be transparent to the user. Although these services are hidden, it does not mean that they are neither useful nor accessible; in contrast, much of their utility will stem from the fact that user need not explicitly issue commands to these hidden computers in order to use available services.

While there have been significant advances in the area of context-aware computing in research environments, we still have to see it making significant inroads in our daily lives. A major prerequisite is the physical infrastructure needed to sense users, their activities, and their environment. Some forms of context might be relatively easy to extract, especially in indoor environments that are already automated to a certain extent. Lighting levels and temperature, for instance, may be obtained from home or building automation systems, or in future home gateways that serve to monitor and control indoor environments. Obtaining other forms of user context, such as the location and orientation of people, may be more challenging. This often involves the deployment of dedicated and often specialized infrastructures of sensors.

A complementary and practical approach might be the integration of sensors as add-on functions on commonly-seen and already-useful devices, services, appliances and infrastructures. Research efforts such as Smart-Its (www.smart-its.org) have focused on providing small everyday devices with some sensing, communication and computational abilities. Schmidt and Van Laerhoven call these smart appliances: devices that are aware of their environment [2]. The core sensing and processing hardware of such devices are often governed by considerations on their mobility, size, weight, and power consumption, and have quite limited computational power. In addition, most of these devices associate with the rest of the computing environment in a wireless and often ad-hoc fashion.

We believe that these small, highly-mobile smart devices will indeed play a critical role in the pervasive computing landscape. However, *smarter* devices that are less mobile and that possess less restrictions on their computing and sensing capabilities will also play an equally important role in providing a ubiquitous communications and computing infrastructure. Aside from sensing the environment, these devices form part of a relatively fixed network infrastructure, participate in distributed computation, and crucially, provide wireless access to their smaller counterparts.

2 Context-Awareness and Adaptation

Much of the research in pervasive and ubiquitous computing has focused on context-aware applications: applications that provide relevant information and services to a user, using information about that user's situation, such as her location, identity, and the state of people, groups and nearby objects [3]. In response to certain situations or changes in context, such an application may automatically alter its execution, trigger the execution of another service, or prompt the user to act in a certain way. We call this type of response *user-oriented adaptation*.

A pervasive computing environment, particularly one that handles diverse forms of media-rich information, also shares many of the issues inherent in the area traditionally known as mobile computing. A user interacting with and moving within such an environment may generate large variations in network traffic due to the diversity of applications being executed, and experience high variability in connectivity characteristics provided by the different access technologies that are available. Since we are interested in smart appliances that provide access to the network, we are equally interested in *network-oriented adaptation*, or the ability of a system to adapt to variability in network conditions such as traffic levels, available bandwidth, congestion levels, and connectivity characteristics [4]. This form of adaptation has been closely studied in systems such as Transend [5], Odyssey [6], Conductor [7], Coda [8], Transformer Tunnels [9], and CM [10].

A valid question would be whether network-oriented context and adaptation are relevant in a pervasive computing environment. The relevance of context always depends on the user's objectives and applications, i.e., her task [3]. In

applications where the quality of the information being transmitted, or the perception of its quality by the user (i.e., fidelity [6]), its timeliness, or the user experience may be affected by variations in the performance of the underlying network, such context is indeed relevant. A fairly sophisticated level of context-awareness in a system would be needed for it to be able to judge whether such context is relevant or otherwise, and to manage both network-oriented and user-oriented adaptation in parallel. To emphasize the level of “smartness” needed to support such context-awareness and adaptation, we call services that enable them *cognitive services* within a pervasive computing system.

3 A Framework for Cognitive Services

Serving as our guide in the design of software that enables cognitive services is a framework that defines the various component functions. A framework can be used as a guide to understanding existing systems, and a useful tool in building new ones. Decoupling the various components of a framework and implementing them separately promotes the design of reusable components with well-defined interfaces, and abstracts the underlying implementation details. Services may be composed from ‘mix-and-match’ combinations of components that satisfy certain criteria for functionality, performance, and resource utilization.

Our goal is to have an integrated framework for both user- and network-oriented adaptation that builds upon a common base for context sensing. We found several useful frameworks for the analysis and design of context-aware and adaptive systems in the literature. One such conceptual framework for network and client adaptation is presented by Badrinath et al. in [11], summarizing the results of several network-adaptive systems. In their framework, adaptation mechanisms are implemented by adaptation agencies, or AAs. An AA consists of an event manager (EM) component that monitors the environment, a resource management and monitor (RM) component that handles resources, and application specific adapters (ASA) that perform adaptation on a data stream.

For context-aware applications, Pascoe identifies four generic capabilities that are needed, namely: contextual sensing, adaptation, resource discovery, and augmentation [12]. In this framework, contextual sensing refers to the detection of environmental states and their subsequent presentation to the user. Applications then adapt their behavior to this contextual knowledge. Contextual augmentation extends these capabilities further by adding information, either through the digital data augmenting reality, or reality augmenting digital data. Contextual resource discovery makes context information and information resources available to interested entities.

Several other models have evolved from work on location systems. WhereMoPS [13] provides a layered system model for indoor geolocation systems that includes data collection, location computation, location normalization, and location provisioning components. In contrast with Pascoe’s approach, WhereMoPS decouples location sensing into separate data collection and computation components, permitting the use of different positioning algorithms. Normalization then

transforms the computed location into a standardized representation, and the information is provided to applications in the provisioning step. The Location Stack [14] model, similarly focused on location context, consists of a seven-layer stack that includes sensors, measurements, fusion, arrangements, contextual fusion, activities and intentions. Although both the WhereMoPS and Location Stack models were originally proposed to handle location context, the essential components may be applied to other types of context sensing.

Dey, Salber and Abowd offer a conceptual framework that includes context widgets, interpreters, aggregators, services and discoverers [15]. Context widgets abstract underlying sensors and acquire context information. This information is further abstracted by interpreters into higher-level information. Aggregators gather information relevant to an entity. Services then execute behaviors using context. Finally, discoverers maintain information on which of these components are available for use by applications.

Schmidt proposes a “perception architecture for context-aware systems,” consisting of sensors, cues, contexts, and the applications that use them [16]. Physical and logical sensors provide information about the world, to be abstracted or processed into symbolic or sub-symbolic values called cues. The context layer then abstracts cues into situations and decides whether a situation satisfies the definition of a particular context. The context is then passed on to applications.

Our framework for cognitive services thus builds upon the functions, characteristics, components and models previously described. It serves as a common framework that integrates network-centered adaptation in mobile computing, and application-centered adaptation in context-aware computing over a unified sensing base. It accounts for the different processes and stages that bridge sensing and adaptation, such as perception, awareness, reasoning, judgment, and augmentation. We impose no strict precedence between these components, and they may be executed in a variety of ways: sequentially, iteratively, or in a recursive fashion:

Sensing. Sensing refers to the collection of measurable or quantifiable physical data or the observation of an event. Such data or events may either be directly measured by hardware or software sensors, or may be higher-level data and events previously detected or generated by other cognitive components and services. Examples of low-level sensed data or events at various levels might be the strength of an 802.11 RF signal, a mobile device associating with an access point, beacon signals impinging on a receiver, the execution of an application, or statistics on the utilization of a network link.

Interpretation. The sensed data are transformed into useful form by interpretation, which may involve the application of a numerical process or an algorithm, by comparing it with a model, or by the application of a logical process such as reasoning. Values obtained may also be checked if they are within expected or acceptable range, and a confidence parameter may be applied to a measurement or estimate. This component may also detect the occurrence of an event based

on the values of data obtained. As with most other frameworks, the output of this stage may be numeric or symbolic.

Augmentation. Sensed data and events may be aggregated with, or examined in relation to, other pieces of information or knowledge. These may be recently-sensed or interpreted data or events originating from other sensors (sensor fusion), or historical data from the same sensor. Historical data may be useful in establishing a trend or in improving the accuracy of estimates from new sensor data through statistical means. Information previously generated may be retrieved from databases, maps, or models. For example, the sensed location of a user may be compared to a map of the location of objects and spaces in order to generate new information on the proximity of users with respect to other entities (“users X and Y are near each other”), or containment within spaces (“user Y is within the space in front of workstation W ”). An important aspect of the augmentation stage is the ability to detect and form relationships and connections between sets of data, events, and prior knowledge.

Adaptation. Adaptation, which we broadly define as goal-oriented action in response to changes in context, may benefit different entities of the system. Some examples we have previously given are user-oriented adaptation and network-oriented adaptation. Although these are by no means the only forms of adaptation that are necessary or present in a pervasive computing system, these are the forms in which we are primarily interested.

The adaptation component in our framework includes the specific actions in response to the detection of certain contexts, or changes in these contexts, as well as the policies and strategies necessary to execute the adaptation itself. Some adaptation strategies include launching new services or applications, modifying application or network behavior, suggesting a course of action to a user, or reserving resources [17]. There may be cases where adaptation may apply even to the components of the framework: sensing, interpreting and augmentation processes may also have to adapt, under certain situations. An interesting form of adaptation that leads us one step closer to truly cognitive services would be *learning*, the ability of a system to modify its view of the world through newly-acquired knowledge.

4 A Platform for Distributed Cognitive Services

We share the view in [17] that motion is part of everyday life and thus a pervasive computing environment should provide mobile and ubiquitous connectivity. Towards this end, we earlier suggested the need for *smarter* appliances that possess not only sensing and distributed computing capabilities, but can also provide connectivity to other devices as well.

Our generalized architecture and physical platform that models and simulates the capabilities of computing appliances in future pervasive environments,

providing computing, connectivity and sensing functions, are called *M-WASPs*, or *Multi-modal Wireless Access and Sensing Platforms*. M-WASPs possess the following basic functions and characteristics:

1. *Access points*. M-WASPs are wireless access points, providing ubiquitous connectivity in a pervasive computing environment. M-WASPs may extend the wireless network by associating in an ad-hoc manner, or by provide wireless access to a wired infrastructure. We also enhance the usefulness of networked appliances and devices by enlisting them to provide wireless access, through embedded M-WASPs.
2. *Multi-modal*. To reflect and support the diversity of connectivity options within pervasive computing environments, M-WASPs are multi-modal, employing a wide variety of connectivity and sensing technologies such as IEEE 802.11, Bluetooth, infrared, and acoustic sensing capabilities. We also explore a wide range of user input devices and classify them as sensors in our framework.
3. *Sensing platforms*. A key requirement for obtaining context is the use of physical and logical sensors for the acquisition of data or detection of events. Sensing may either be a dedicated function for a device or a software program, or it may be an add-on or overlay function. For example, in RADAR [18], 802.11 wireless access points were used beyond their traditional function of providing connectivity, to sense user locations. We intend to exploit this fully using other interfaces and technologies.
4. *Programmable*. A key feature of the M-WASP architecture is programmability in the active networking context [19],[20]. An active networking approach allows us to flexibly deploy cognitive service components within the network on an on-demand basis. We push the active networking paradigm further by supporting not only mobile code, but mobile data as well, within our wireless access points.

Active networks have long been recognized as a promising infrastructural solution for adaptation [7],[11]. The Mobiware middleware toolkit [21], for example, is based on an open programmable networking architecture and runs on mobile devices, wireless access points and mobile-capable switches and routers. This effort has produced a useful set of objects and APIs to provide QoS support for adaptive mobile networking.

In application-level active networks (ALAN), active networking principles are applied at the application layer rather than the lower layers of the network. FunnelWeb [22] is an ALAN implementation that supports the dynamic deployment and execution of proxylets in active nodes in the network. These proxylets implement network-oriented adaptive functions such as transcoding, compression, and caching. We have previously used the ALAN approach for the segmented adaptation of traffic aggregates [23] and to dynamically manage and optimize peer-to-peer traffic on bandwidth-limited links [24].

In real deployments, M-WASPs may represent a wide range of devices: infrastructure elements such as routers and dedicated wireless access points, mobile

user appliances such as PDAs and mobile phones, or computing elements embedded in a diverse range of appliances such as office equipment, networked environmental control systems, pieces of furniture, and household appliances. At the extreme end, M-WASPs may be dedicated computing devices such as PCs. M-WASPs may vary in terms of actual capabilities and configurations such as processing power, memory and storage, interfaces, connectivity and sensors, according to their intended application. However, the heterogeneity of the underlying hardware is abstracted by an architecture whose primary function of interest from our viewpoint will be the provision of computing, connectivity and sensing functions – distributed cognitive functions – in addition to any general or specialized embedded computing function the particular device might serve.

5 Sensing and Representing Location Context in M-WASPs: Examples

M-WASPs are *smart wireless access points* in the sense that in addition to providing wireless access, they are also actively involved in providing distributed cognitive services, such as adaptation functions at the user and network levels. Sensing various forms of user and system context is also another main function. While the notion of “context” that may be relevant to a user may involve a wide range of things, the location of users and surrounding objects plays a very important role and consequently has been the subject of much interest for many researchers.

In this section we discuss some of the schemes we use in M-WASPs to sense and represent fine-grained location information in an experimental setting.

5.1 A Method for Sensing Location

For our experimental use, we needed a location sensing scheme that would be compatible with commercially-available off-the-shelf devices such as a PDA. To roughly estimate the position of users within a relatively large area, such as within a 3–4 meter radius, techniques such RADAR [18] and its variants may be used with a PDA outfitted with an IEEE 802.11 interface. For fine-grained positioning, such as within 10–20 cm., while a number of systems such as the Active Bat Location System [25] and Cricket [26] have been discussed in the literature, most of these systems usually use ultrasonic transducers and a channel for transmitter-receiver synchronization such as an RF or infrared channel. We preferred a position sensing scheme that would use available interfaces and require only a bare minimum of hardware interfacing, if any at all.

Aside from our preference in using off-the-shelf components and built-in interfaces, we likewise made an assumption that the located object would operate asynchronously with respect to the positioning system. A location estimation technique known as hyperbolic multilateration does not require the tracked object to be synchronized with the positioning system. If beacons, or pairs of beacons can be closely time-synchronized with each other, a receiver could detect the

arrivals of each beacon's signal and measure the relative delays between them. If signals from beacons i and j arrive at a receiver at t_i and t_j , respectively, referenced to the receiver's clock, then the time-difference-of-arrival (TDOA) is simply $t_i - t_j$. For a receiver located at coordinates (x, y, z) , and any two beacons i and j located at (x_i, y_i, z_i) and (x_j, y_j, z_j) respectively, the equation describing the range difference r_k corresponding to the TDOA $t_i - t_j$ for this pair is given by:

$$\begin{aligned} r_k &= c(t_i - t_j) \\ &= \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \end{aligned} \quad (1)$$

where c is the propagation speed of the beacon signal used. Equation (1) describes a hyperboloid, and with at least four beacon signals, three independent TDOA values may be obtained, producing three independent equations. The solution to these three simultaneous equations yields the (x, y, z) position estimate of the receiver. Graphically, this corresponds to the intersection between the hyperboloids generated using (1) for any set of three TDOA values.

We applied this technique using acoustic spread-spectrum signals, with standard PC speakers as beacons and a receiving microphone as the located object. The use of spread-spectrum facilitates the detection of the arrivals of beacon signals at the receiver due to their excellent correlation properties, and provides some measure of resilience against noise and environmental scattering [27]. Four PC speakers, each with a single 3.5-inch driver, were positioned in a room. Three of these were mounted on the ceiling, approximately 2.2m above the floor on average, while one was mounted on a wall, approximately 80.5 cm above the floor level. Acoustic beacon signals, consisting of 127-bit Gold codes with a chip rate of 10 kchips/s and BPSK-modulated with a 10 kHz sine wave, were simultaneously transmitted through the speakers. A microphone (simulating a PDA in our test scenario) recorded the received signal every 10 cm on a 130 cm x 110 cm grid. The recorded signal was then successively correlated with each of the transmitted Gold codes. A correlation peak indicated the instant that a beacon signal arrived at the microphone. The speed of sound is approximated to the first order using the formula

$$c \approx 331.5 + 0.610t_{\text{air}} \quad (2)$$

where t_{air} is the air temperature in degrees Celsius and c is in meters/second. In our experiments, the temperature was recorded from a digital thermometer. In an actual implementation, the ambient temperature may be supplied by an online sensor. Alternatively, a fixed approximate value may be used in environments where the temperature is regulated or typically does not vary to a large degree. Since the equations for each TDOA pair represented by (1) are nonlinear, we linearized them using the first two terms of their Taylor series, and used least squares to solve the resulting equations.

The results of one of the trials of our acoustic position sensing scheme is shown in Fig. 1. The positions marked with "x" indicate the actual microphone

positions, while positions marked with “o” indicate the position estimates computed through hyperbolic multilateration. Lines interconnect pairs of actual and computed positions. The gaps in the grid where there are no “x” marks represent points where the least-squares algorithm did not converge within the maximum number of iterations, or the resulting computed position was outside the coordinate system. For the data shown in the figure, the computed position deviated from the actual position by 7.0 cm on average, and 80% of all computed positions deviated by less than 9.4 cm from their actual positions. In sensing the location of people and objects, we are normally more interested in their (x, y) position rather than their elevation above the floor. For the data represented in Fig. 1, the deviation from the actual positions along the $x - y$ plane was around 4.6 cm on average, and less than 7.5 cm for 90% of all computed positions.

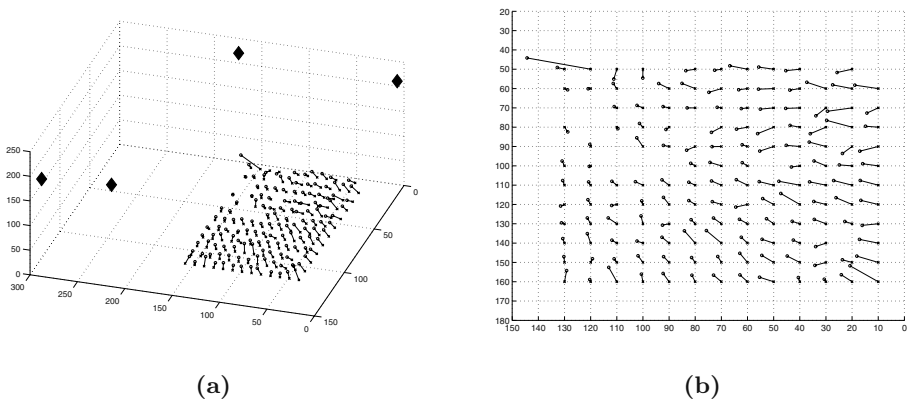


Fig. 1. (a) 3D plot showing measurement points, computed positions and beacon locations. (b) Top view. (Axes are marked in centimeters.)

In an actual implementation the position may be computed on the PDA itself by performing the hyperbolic multilateration computations on the sensed beacon signals and a preloaded database of beacon codes and the corresponding (x, y, z) coordinates of the speakers transmitting them. Alternatively, the position information of the speakers may be contained within the beacon signals, although this would require more processing on the part of the PDA and the use of longer code sequences. Either way, the PDA’s location would not be known to the network, so this may be considered a mode of operation that preserves the privacy of the user. A non-privacy or tracking mode would have the PDA transmit the recorded beacon signals back to the M-WASP, using either an 802.11 or Bluetooth interface, for the location computations. This mode sacrifices privacy in favor of hardware simplicity, as it shifts the computational burden from the tracked object to the network. The system we have described can be rapidly deployed and used, as no additional hardware construction is

needed. Our technique does not require any synchronization between the beacons and the located object, and allows very simple commercial devices with little or no computational power, such as an analog wireless microphone, to be tracked. Even an ordinary audio recorder, for example, may continuously record acoustic beacons as it moves within an area, and its traversed path may later be post-processed and reconstructed. However, in a real-life scenario, rather than a laboratory deployment, the use of audio beacons in the audible range might be annoying to users. In such cases, it would be necessary to shift the working frequencies to the ultrasonic range, and although this would require some simple hardware modifications, the basic principles would remain the same.

5.2 Representing Spaces

For location context to be useful to applications, information on the location of users, nearby objects and spaces of interest must be represented in a form that can easily be stored, transmitted and processed. Having a simple and efficient scheme for representing location and spaces, and other forms of context in general, is desirable for the following reasons:

Support for simple but smart devices. A scheme that is simple enough such that context may be exchanged, stored and processed by simple devices with limited memory and processing capacity promotes a distributed model of context-awareness and cognition. Such a model not only supports users who ask “What useful objects or services are nearby?” but also allows simple devices to pose the similar question “Who are the nearby potential users of my service?”

Robustness. If context can be distributed throughout the system and processed in a distributed manner, then there is less chance of catastrophic failure, in contrast with a system that relies on a centralized server to process and store location information.

Privacy. Privacy can be enhanced if the user can select the entities that will be made aware of her context. In some cases, it may be sufficient to share context locally to nearby devices on a need-to-know basis. This is possible if these nearby smart devices can process context information locally within the constraints of their limited computing and storage resources.

Network-friendliness. A scheme that uses a simple representation minimizes the bandwidth consumed by the exchange of location context, a crucial consideration for wireless networks.

We now discuss a scheme for representing locations and spaces that satisfies our design requirements for simplicity and efficiency.

Large spaces such as rooms, are partitioned into small cubes, and each unit cube is uniquely identified by a set of Gray-coded coordinates, similar to Karnaugh maps [28]. A more precise term used in Boolean algebra for these unit cubes are 0-cubes. A 0-cube may be contained within one or more spaces of

interest, such as “the space in front of workstation W ,” and such a space of interest is completely defined by the set of cubes that completely enclose it. This is similar to the definition of space containment in [25]. Spaces of interest may thus be represented in terms of Boolean functions that describe the logical sums of 0-cubes that completely enclose them. With spaces represented by Boolean functions, various logic operations may then be applied to determine relationships between spaces and the location of objects and users, such as intersection and containment. Operations such as combining spaces may be done through a logical union of Boolean expressions. Proximity may be determined by testing for inclusion within larger enclosing spaces.

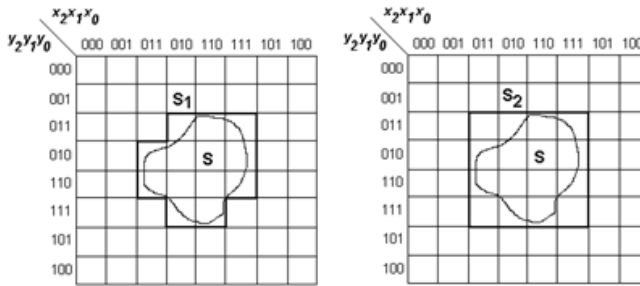


Fig. 2. A space of interest S and enclosing spaces S_1 and S_2 are drawn on a Karnaugh map

Most of the spaces in which we are interested, e.g. “the space in front of workstation W ” consist of collections of contiguous groups of unit cubes. We exploit this property to obtain compact space representations by applying logic minimization techniques. This approach is similar to the representation and compression of images in [29], however, our goal is not to compress an image of the entire area including the details of all objects contained within. Rather, our goal is to obtain simple and compact representations only of the spaces and locations of interest. A two-dimensional simplified example of this space representation scheme is shown in Fig. 2. A space of interest, S , is shown on this figure. We also indicate two areas, S_1 and S_2 , that completely enclose S . The Boolean expressions for S_1 and S_2 in minimal sum-of-products form are

$$f_{S_1} = x_1y_1y'_0 + x_1x'_0y_1 + x_2x_1y'_2y_1 \quad (3)$$

$$f_{S_2} = x_1y_1 \quad (4)$$

It can be seen from this example that the geometry of the enclosing area used to approximate the original space of interest affects the extent to which the switching expression can be minimized. Generally, in building the maps that represent our spaces and the objects contained, we can select geometries that will lead to more compact representations. One advantage of this space representation

scheme is that it implicitly encodes both position and size information. However, there are some potential weaknesses. First, as previously mentioned, space geometries other than those that may be conveniently enclosed in rectangular cubes with dimensions that are integer powers of 2 may tend to have more complex expressions. Second, it tends to view the world in “black and white,” that is, if the location of an object is represented by Boolean expression f_{S^*} , then the representation implies that the probability of actually finding the object outside S^* is zero. Third, within S^* , the probability of finding the object is uniformly distributed. These might not be the case if the area being represented by f_{S^*} is generated by a sensor that inherently can only provide an estimate of location.

One solution to the first problem above, aside from careful geometry selection, is to explore other shapes that lead to simple expressions in other minimal two-level Boolean representations such as exclusive-OR sum of product (ESOP) forms. Additionally, if the space information is to be shared, the simpler product terms in the minimized expression (corresponding to larger and more regular sub-areas) can be transmitted first for rapid, “short-circuit” logic evaluation. In some cases this may eliminate the need to transmit the other product terms, and thus while the stored representation remains complex, it does not generate too much traffic. Finally, to represent probability distributions that are non-uniform, a space may be built from composite overlapping or non-overlapping sub-areas and a probability assigned to each component.

6 Related Work

Girod and Estrin use acoustic spread-spectrum techniques in a ranging system that uses frequencies in the audible range [27]. Although their system only produces range (distance) information rather than location, it may be extended through multilateration in order to estimate location. This has in fact served one of the technical bases for the acoustic location sensing component we have developed for M-WASPs. In addition, their philosophy of using COTS hardware has served as a guide for us in designing a system that can be rapidly deployed and used in conjunction with commercial devices such as PDAs.

A privacy-oriented location system based on ultrasonic DS/CDMA spread-spectrum and pseudoranging has recently been presented by Hazas and Ward [30]. Although the physical sensing base is identical to ours, the difference in approach (hyperbolic multilateration vs. pseudoranging), i.e., the interpretation component, has some implications in the overall design of the location system, particularly in the need for synchronization within the system. A pseudoranging system typically requires beacons to be tightly synchronized with each other, and there is likewise some benefit in synchronizing the located object with the beacons as well, as this minimizes the magnitude of the clock bias that needs to be estimated. In a hyperbolic multilateration system similar to ours, tight synchronization is required only among pairs of beacons to provide accurate TDOAs, and to a lesser extent, across different pairs. While synchronization between the beacons and the located object may minimize the number of acoustic

data samples that need to be processed, the algorithm itself does not require it. At any rate, our ability to use a different interpretation component, i.e., post-processing algorithm, over the same physical sensing base as may be dictated by the sensor deployment scheme or other system concerns, precisely illustrates the usefulness of the modular framework we have presented.

Broadly, while our research shares a number of common objectives with similar research efforts in context-awareness within the pervasive and ubiquitous computing research community, we extend these further by paying equal attention to the problems caused by the mobility of users and devices, as well as their information flows, within these environments. Our work is more related to efforts such as Carnegie Mellon's Aura [31], which seeks to create distraction-free environments that adapt to users' context and needs. Aura applies the concepts of proactivity, or the ability to anticipate requests from a higher layer, and self-tuning, or autonomous adaptation by layers. Unlike many similar efforts in the ubiquitous computing domain, Aura correctly provides attention to the network-oriented adaptation that is required to support user mobility. We seek to achieve similar broad objectives through our application of a well-defined framework for cognitive services, delivered within the pervasive computing environment in a distributed, scalable and robust fashion, through our programmable, smart wireless access points called M-WASPs.

MIT's Project Oxygen (<http://oxygen.lcs.mit.edu>) aims to provide human-centered computation freely available everywhere. Environmental devices, called E21s, provide sensing, computational and communication functionality for intelligent spaces. These devices are interconnected through flexible, decentralized, adaptive networks called N21s. Our M-WASPs seem to share characteristics similar to E21s, and our active networking approach allows the deployment of services necessary to manage networks similar to N21s within the same platform, using a integrated framework for cognitive services.

7 Conclusions

We have introduced M-WASPs, which are smart wireless access points for pervasive computing. M-WASPs provide a wireless infrastructure linking users with smart devices and other networks. They also provide sensing and cognitive functions, enabling user- and network-oriented adaptation. We have also presented a framework for distributed cognitive services that includes components for sensing, interpretation, augmentation and adaptation. This framework serves as our guide in designing the active code that can be dynamically deployed and executed on M-WASPs. Having efficient and simple means of sensing, representing, exchanging and processing context, such as space information, promotes a distributed, modular approach to cognitive services. As an example, we presented a scheme we use in M-WASPs to sense and represent location and spaces. These simple techniques for sensing and representing location and spaces may be of interest for experimenters in context-aware computing and in related domains where positioning may be needed, such as in robotics and virtual reality.

Acknowledgment. Roel Ocampo's work is supported by a Doctoral Studies Fellowship from the University of the Philippines.

References

1. M. Weiser. The Computer for the 21st Century. *Scientific American*, September 1991.
2. A. Schmidt and K. Van Laerhoven. How to Build Smart Appliances? *IEEE Personal Communications*, 8(4), August 2001.
3. A. K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing Journal*, 5(1), 2001.
4. R. Katz. Adaptation and Mobility in Wireless Information Systems. *IEEE Personal Communications*, 1(1), 1994.
5. A. Fox, S. D. Gribble, Y. Chawathe and E. A. Brewer. Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives. *Proc. 16th Intl. Symposium on Operating Systems Principles (SOSP-16)*, France, October 1997.
6. B. Noble, M. Satyanarayanan, D. Narayanan, J. Tilton, J. Flinn and K. Walker. Agile Application-Aware Adaptation for Mobility. *Symposium on Operating System Principles*, November 1997.
7. M. Yarvis, P. Reiher and G. Popek. Conductor: A Framework for Distributed Adaptation. *Proc. 7th Workshop on Hot Topics in Operating Systems*, March 1999.
8. L. B. Mummert. *Exploiting Weak Connectivity in a Distributed File System*. PhD thesis, Carnegie Mellon University, School of Computer Science, 1993.
9. P. Sudame and B. R. Badrinath. Transformer Tunnels: A Framework for Providing Route-Specific Adaptations. *USENIX Annual Technical Conference*, June 1998.
10. D. Andersen, D. Bansal, D. Curtis, S. Seshan, and H. Balakrishnan. System Support for Bandwidth Management and Content Adaptation in Internet Applications. *Proc. 4th Symposium on Operating Systems Design and Implementation*, San Diego, CA, October 2000.
11. B. Badrinath, A. Fox, L. Kleinrock, G. Popek, P. Reiher, and M. Satyanarayanan. A Conceptual Framework for Network and Client Adaptation. *IEEE Mobile Networks and Applications*, 5(4), December 2000.
12. J. Pascoe. Adding Generic Contextual Capabilities to Wearable Computers. *Proc. 2nd International Symposium on Wearable Computers*, October 1998.
13. M. Wallbaum. WhereMoPS: An Indoor Geolocation System. *The 13th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications*, Lisbon, Portugal, September 5-18, 2002.
14. J. Hightower, B. Brumitt, and G. Borriello. The Location Stack: A Layered Model for Location in Ubiquitous Computing. *Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002)*, Callicoon, NY, June 2002.
15. A. K. Dey, D. Salber, and G. D. Abowd. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction (HCI) Journal*, 16 (2-4), 2001.
16. A. Schmidt. *Ubiquitous Computing – Computing in Context*. PhD dissertation, Computing Department, Lancaster University, November 2002.
17. M. Satyanarayanan. Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, 8(4), August 2001.
18. P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location And Tracking System. *Proc. IEEE Infocom 2000*, Tel-Aviv, Israel, vol. 2, March 2000.

19. D. L. Tennenhouse and D. J. Wetherall. Towards an Active Network Architecture. *Computer Communication Review*, 26(2), April 1996.
20. A. T. Campbell, H. G. De Meer, M. E. Kounavis, K. Miki, J. B. Vicente and D. Villela. A Survey of Programmable Networks. *ACM SIGCOMM Computer Communication Review*, 24(2), April 1999.
21. A.T. Campbell, M.E. Kounavis and R. R.-F. Liao. Programmable Mobile Networks. *Computer Networks*, 31(7–8), April 1999.
22. M. Fry and A. Ghosh. Application Level Active Networking. *Computer Networks*, 31(7), April 1999.
23. H. De Meer and P. O’Hanlon. Segmented Adaptation of Traffic Aggregates. *Proc. Quality of Service – IWQoS 2001, 9th International Workshop*, Karlsruhe, Germany, June 2001.
24. H. De Meer, K. Tutschku, and P. Tran-Gia. Dynamic Operation in Peer-to-Peer Overlay Networks. *Praxis der Informationsverarbeitung und Kommunikation (PIK Magazine), Special Issue on Peer-to-Peer Systems*, June 2003.
25. A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The Anatomy of a Context-Aware Application. *Proc. 5th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MOBICOM)*, Seattle, WA., August 1999.
26. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. *Proc. 6th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MOBICOM)*, August 2000.
27. L. Girod and D. Estrin. Robust Range Estimation Using Acoustic and Multimodal Sensing. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, Maui, Hawaii, October 2001.
28. M. Karnaugh. The Map Method for Synthesis of Combinational Logic Circuits. *Trans. AIEE. pt I*, 72(9):593–599, November 1953.
29. A. K. Chaudhary, J. Augustine, and J. Jacob. Lossless Compression of Images Using Logic Minimization. *Proc. Intl. Conf. on Image Processing*, Vol. 1, September 1996.
30. M. Hazas and A. Ward. A High-Performance Privacy-Oriented Location System. *Proc. First IEEE International Conference on Pervasive Computing and Communications (PERCOM ’03)*, Dallas-Fort Worth, USA, March 2003.
31. D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste. Project Aura: Toward Distraction-Free Pervasive Computing. *IEEE Pervasive Computing*, April-June 2002.

NAT-Based Internet Connectivity for On-Demand Ad Hoc Networks

Paal Engelstad and Geir Egeland

Telenor R&D,
1331 Fornebu, Norway
{Paal.Engelstad, Geir.Egeland}@telenor.com

Abstract. A prerequisite for widespread and successful deployment of on-demand ad-hoc networking technology is its ability to provide easy access to the Internet. Existing solutions for Internet access are mainly based on modifying Mobile IPv4 (MIPv4). An easier approach, yet poorly documented in published material, is to implement Network Address Translation (NAT) on Internet Gateway nodes in the ad hoc network. In this paper we describe problems experienced in our lab test-bed with NAT-based solutions under the condition of site multi-homing. Based on this experience, we propose a working solution for multi-homed ad hoc networks.

1 Introduction

IP-based applications, such as web browsing, e-mail, telnet and ftp, mainly communicate with servers or peers over the Internet. A mobile ad-hoc network (MANET [1]) has no fixed infrastructure and services on the Internet might not be available in these networks. A likely scenario is that nodes on an ad-hoc network in some cases also want to connect to nodes on the Internet, using services available here. For a widespread and successful deployment of MANETs, the ability to provide easy access to the Internet is therefore a prerequisite.

A possible solution is to let a node that is participating in a MANET operate as an Internet gateway and provide other nodes on the MANET with Internet access. One approach is to implement a Mobile IPv4 Foreign Agent (MIP-FA) on the gateway ([2], [3]). MANET nodes that require Internet access, implement a Mobile IPv4 (MIPv4 [4]) client, and register the globally routable IPv4 address of the gateway as a care-of-address with their MIPv4 Home Agents (HA). However, as we describe in this paper, a Mobile IP-based solution has a number of drawbacks and introduces high complexity to implementations.

An easier way to provide IPv4 Internet access to MANET nodes is to implement Network Address Translation (NAT) on the gateways [5]. Although NAT solutions have emerged in different test-bed implementations (see for example [6]) little has been documented, neither in scientific papers nor in IETF Internet Drafts.

A challenging issue with the use of NAT in general relates to site multi-homing. Since a NAT-device translates both outgoing and incoming packets, all

traffic belonging to one communication session (e.g. TCP session) must traverse through the same NAT-device, otherwise the session will break. When the site is multi-homed, it can be difficult to control that all packets from one session are consistently routed through the same NAT-device.

This issue is also highly relevant to MANETs. A MANET is without infrastructure, and it is difficult to eliminate the possibility of multi-homing. It is not possible to control the network behavior in such a way that there is only one node on the MANET operating as a MANET Internet gateway. Even if there existed a simple solution to suppress a second node to operate as a gateway, the problem would re-emerge at the moment when two MANETs, each with a NAT-based gateway, merge into one network. Shutting down one of the gateways would mean to break all on-going communication sessions over that gateway.

In this paper we address the lack of a good mechanism for IPv4 Internet access in on-demand MANETs, i.e. a MANET that is routed with a reactive routing protocol, such as the Ad hoc On-demand Distance Vector (AODV [7]) routing protocol or the Dynamic Source Routing (DSR [8]) protocol. The paper examines the use of NAT for this purpose and point out potential problems with NAT-based solutions and multi-homing as experienced in our test-bed. Based on this, a working solution for multi-homing is proposed.

In Section 2 we present proposed solutions for providing Internet connectivity on on-demand MANETs. In Section 3 we analyze how a NAT-based solution work in scenarios of multi-homing. A proposed solution for NAT-based gateways based on our findings is presented in Section 4. The applicability of our findings to MIP-FA-based solutions is also discussed in Section 5.

The main focus is set on on-demand networks that use AODV as a routing protocol. However, as the analyses and proposals presented in this paper are of a general nature and not strictly dependent of the reactive routing protocol, we will deal with applicability of our findings to DSR in a separate section by the end of the paper.

2 Background

2.1 Reactive (“On-Demand”) Routing Protocols

A number of reactive routing protocols have been proposed. The most widely studied and popular proposals include the AODV and the DSR protocols.

Reactive protocols allow source nodes to discover routes to a destination IP-address on demand. Most proposals, including AODV and DSR, work as follows: When a source router needs a route to a destination for which it does not already have a route, it issues a Route Request (RREQ) packet. The packet is broadcasted by controlled flooding throughout the network, and sets up a return route to the source.

If a router receiving the RREQ is either the destination or has a valid route to the destination, it unicasts a Route Reply (RREP) packet back to the source along the reverse route. The RREP normally sets up a forward route. Thus, the

RREQ and RREP messages set up two uni-directional unicast routes in opposite directions between source and destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. (The acronyms RREQ and RREP are borrowed from AODV.)

Different reactive routing protocols have different strategies to deal with route maintenance and route repair. Most protocols let routes that are inactive eventually time out. If a link break occurs while the route is active, the routing protocol normally implements an algorithm to repair the route.

Different protocols have different ways to manage routing state information. With AODV, for example, correct forwarding of packets between source and destination relies on stored state information in routing tables on intermediate nodes in the network. With DSR, on the other hand, the sender of the packet encodes the entire route explicitly into the packet, and the packet is source-routed from source to destination.

2.2 Basic Functionality for Internet Connectivity

Providing Internet access to a MANET node – hereafter referred to as a “Source Node” (SN) – via another MANET node that has direct access to the Internet – hereafter referred to as a “gateway” (GW) – involves a number of steps.

Before packets from the SN are routed to a node outside of the MANET, one has to determine somehow that the destined node is not present on the MANET. If the destination IP address is not found on the MANET, available gateways must be detected, and one of these must be selected.

The selected gateway must somehow provide a source IP address to SN’s outgoing traffic to make sure that the incoming traffic returning from the Internet is correctly routed back to the gateway. Due to the shortage of globally routable IPv4 addresses, an Internet gateway is likely to be assigned only one IP-address on the external network to which it is connected. Hence, all MANET nodes that use this gateway to access the Internet must share the external IP address of the gateway for traffic returned from the Internet.

Some explicit signaling between the SN and the gateway might be required (e.g. for a MIP-FA-based solution), or implicit translation at the gateway is required (e.g. for a NAT-based solution). Finally, as an integrated part of the previous steps, or as a separate step, there must be established forward and return routes between the SN and the gateway.

In summary, the basic functionality required for Internet connectivity includes:

- Determining that an address is not present on the MANET
- Detection of different available gateways
- Selection of one gateway
- Signaling with selected gateway (if required)
- Providing a globally routable IP address for incoming traffic
- Routing of outgoing traffic via gateway
- Routing of incoming traffic via gateway

In the following, a MIP-FA-based solution and a NAT-based solution are presented and described in relation to this list of required functionality.

2.3 MIP-FA for AODV

There exist at least two proposals that attempt to implement a MIP-FA-module on a gateway for AODV. The MIPMANET proposal [3] takes an academic approach and proposes a number of schemes to optimize the use of Mobile IPv4 for this purpose. The scheme proposed by Belding-Royer *et. al.* [2] represents a similar, but considerably simpler engineering approach. We will discuss the latter approach, since this proposal is more relevant to the analyses undertaken in this paper.

A MIP-FA-based gateway will periodically flood Agent Advertisements throughout the MANET. From this other nodes in the MANET will learn the IP address of the gateway and the care-of-address that it offers. The SN can select a gateway and use AODV mechanisms to discover a route to it. Once the route is known, the SN unicasts a Registration Request message to the gateway, thereby registering with the FA-module on the gateway and SN's own HA.

When the SN searches for a node, it sends a RREQ, whether the node is present locally on the MANET or not. Upon reception of a RREQ message for a destination that the gateway believes is present outside the MANET, the gateway transmits a RREP to the requesting node with an 'F'-bit set to indicate that the destination node can be reached through the gateway.

The destination IP-address in the RREP is set to the IP-address of the node the SN is searching for. This means that the gateway sends out "Proxy RREPs" on behalf of nodes that might be present on the Internet. The advantage of this approach is that the SN can send packets to external nodes in the same way as it would to nodes that are present on the MANET.

A mechanism is required to ensure that communication will not escape through a gateway when the destined node is present on the MANET. This can be solved by ensuring that the destination sequence number in an RREP from a MANET node present on the network is always larger than an that of an RREP from a MANET gateway, when the two RREPs are triggered by the same RREQ. Since the highest sequence number gives preference during route discovery, routes to MANET nodes present on the MANET will always have preference over routes established by Proxy RREPs sent by gateways.

One solution is to mandate that a MANET node that is present on the MANET (while not operating as a gateway) must update its destination sequence number before issuing the RREP in response to an RREQ with the F-flag set. The MIP-FA-based gateway, on the other hand, will always copy the AODV-specific destination sequence number of the RREQ into the corresponding field of the RREP. Another solution is to mandate that SN always set the Unknown Sequence Number flag to 1 and the destination sequence number to zero in RREQs with the F-flag set. Then the RREPs from MANET gateways will always have a zero destination sequence number.

2.4 AODV-UU NAT Solution

Uppsala University's implementation of AODV [6] includes a NAT-solution. Mobile hosts are unaware of its presence, hence there is no NAT discovery and NAT uses Proxy RREP to reply to RREPs destined for hosts on the Internet.

Unlike the MIP-FA solution, the gateway uses its own AODV destination sequence number when replying with Proxy RREPs. This may potentially introduce a problem since a route will always be set up to the node that sends the RREP with the highest sequence number. Hence, in cases where the destination sequence number of the gateway is higher than that of the destination node, packets will be routed out through a gateway even if the destination node is present on the MANET.

To prevent this, the AODV-UU NAT solution mandates that all addresses on the MANET belong to the same subnet prefix. Hence a NAT will reply with a Proxy RREP only if the destination IP address is not under this prefix.

This prefix limitation conflicts with the widely accepted notion that a node should be able to bring any address into the MANET, and use this while present on the network. The prefix limitation might also introduce problems when other gateway technologies are present on the network. One example is that the NAT-based gateway and a MIP-FA-based gateway are mutually exclusive, since Mobile IP mandates that mobile nodes use their home address on visited links, i.e. an address that not necessarily belongs to a prefix assigned to the MANET.

We did not find the AODV-UU NAT solution satisfactory, so we implemented an alternative approach to the prefix limitation. The NAT was allowed to answer to all RREQs, similar to the MIP-FA-based solution of Belding-Royer et al. [2]. A targeted node would have to increase its destination sequence number before replying with an RREP, while a MANET gateway would copy it from the destination sequence number field of the RREQ. Hence, if the targeted destination address were present on the MANET, a direct route to that node would have preference, since the RREP from the targeted node would have a larger destination sequence number.

In our further references to NAT-based solutions, we refer to solutions without the aforementioned prefix limitation.

2.5 Comparison of MIP-FA-Based and NAT-Based Solutions

A drawback with a MIP-FA solution was revealed after implementing it in our research lab. The scheme requires changes to the Mobile IP implementation on both the Mobile Host (MH) side (i.e. on the SN requiring Internet access) and on the Foreign Agent (FA) side (i.e. on the gateway). Since the MH and the FA are no longer on-link, both sides will have to deal with Agent Solicitations and Agent Advertisements in a different way; TTL values and IP destination addresses must be set differently; ARP must be used differently and MAC-addresses are no longer relevant for communication between MH and FA. Moreover, independent co-existing implementations of MIPv4 and a MANET routing protocol

are not trivially managed, since both implementations will make unsynchronized modifications to the routing table.

Another drawback that limits the applicability of a MIP-FA based solution is that it assumes that the care-of-address of the gateway is globally routable. However, the IPv4 address space is a scarce resource, and many MANET gateways might only be able to acquire a private IPv4 address on the external network to which they are connected.

A NAT-based mechanism, on the other hand, appears as a considerably easier solution. The NAT functionality may be in the form of Basic NAT, however NAPT (i.e. NAT with port translation) is a more applicable solution, since many MANET Internet Gateways might only be able to acquire a single IP-address on the external network to which they are connected [5]. NAT-devices can be nested, and this solution will work even when the MANET Internet Gateway acquires a private IP address from the external network.

In the next section we take a closer look at problems arising with NAT-based solutions, especially when the MANET is multi-homed.

3 Problems with NAT and Multi-homing

3.1 NAT-Based Multi-homing with One Source Node

We tested site multi-homing in an on-demand MANET with two NAT-based gateways present. The source node (SN) communicated with an external host (XH), and both gateways (GW1 and GW2) were reachable through the same Intermediate Node (IN). The AODV-UU implementation for Linux [6] was used as routing modules on all MANET nodes (i.e. SN, IN, GW1 and GW2), and WLAN 802.11b was used for the wireless communication. All nodes were located in the same room (9-by-9 meters), and MACKILL [6] was used to emulate that SN was not in direct radio-range with the gateways. The test configuration is illustrated in Figure 1.

Initially the SN established a route to XH over one of the gateways, i.e. GW1. The SN then established a TCP connection with XH, and periodically sent over short messages on 1 seconds intervals. Using the parameters proposed by the AODV specification, the route between the SN and GW1 would time out after 3 seconds. However, since AODV uses the data traffic to update routes, the route would never time out before the TCP connection sent a new packet. Hence, the route remained active, stable and unaltered without experiencing any significant problems.

As we increased the transmission interval of data packets to a period of 4 seconds, however, we experience serious problems. When a new packet was to be sent, the route had already timed out and SN had to discover a new route to the XH over GW1. Due to dynamics in the system, sometimes the RREP from the other gateway (GW2) would be the first to arrive at the IN. In these cases, the IN established a route to the XH through GW2.

This happens since the RREPs from both gateways carry the same destination sequence number, because both copy it from the RREQ sent by the SN.

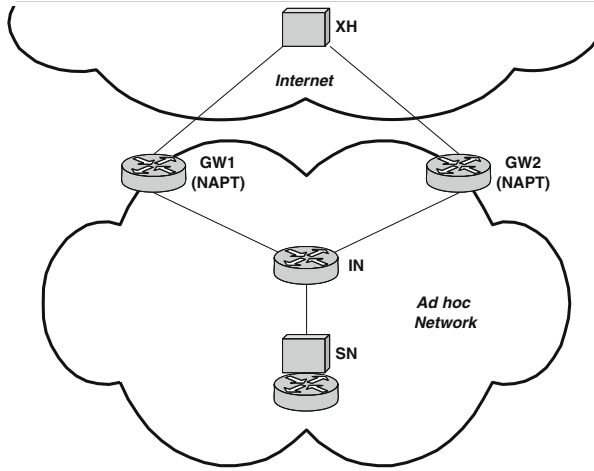


Fig. 1. Test-bed implementation with one Source Node (SN) communicating with an external host (XH) over the Internet. There are two NAT-based gateways (GW1 and GW2) present on the MANET, and all communication passes through an Intermediate Node (IN).

Furthermore, both RREPs carry the same hop-count since both gateways are one hop away from IN. Hence, when the RREP from GW1 finally arrived approximately a millisecond later, the IN did not update its route for the XH via GW1. According to the AODV specification, the IN has to discard and not process RREPs for a valid route unless its destination sequence number is higher or its hop-count is lower than those of the RREP that established the route. (Figure 2.)

When the outgoing TCP packets passed out through the NAT-module of GW2, the module naturally translated the source IP address of the TCP packets to a different address than the one used by the NAT-module at GW1. The packets were not recognized when they finally arrived at XH, and the TCP session therefore broke.

We experienced that 11% of the time the RREPs from GW2 were the first to arrive at the IN and thus established the route through GW2, whereby the TCP session would break due to the route change. Sessions were able to send only approximately 10 packets on average before they would break.

When the route changes, the XH will normally respond by a “TCP Reset” message, to which different applications will react differently. When we tested Telnet [9], for example, the application would shut down the Telnet connection immediately upon reception of the “TCP Reset” message.

There will always be a certain amount of non-deterministic dynamics in wireless ad-hoc networks, due to factors such as radio fading, node mobility, packet collisions and so forth. The two gateways will also have different performance due to the internal states of the operating systems. To study these effects un-

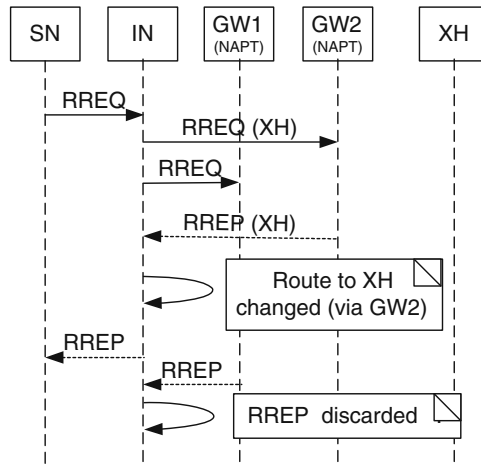


Fig. 2. A sequence diagram describing how the race conditions occur in the test-bed configuration described in Figure 1.

der more controllable conditions we emulated varying transmission times over the links by allowing the gateways to delay the transmission of outgoing Proxy RREP packets in response to RREQ requests for the XH. Each Proxy RREP was delayed for an arbitrary time between zero and T_{max} ms. (The possible varying state of the link between SN and IN was not considered significant in this trial.)

For each chosen value of T_{max} , we performed a series of 40 trials, each consisting of 400 RREQs sent with a packet transmission interval of 4 seconds. For each series of trials we incremented T_{max} from 0 to 5 ms. Results are presented in Figure 3. It shows that $T_{max} = 0$ corresponds to the case where only 11% of the RREPs from GW2 were the first to arrive at the IN. As T_{max} increases the introduced variation begins to dominate over the test-bed-specific variations, and the ratio of RREPs from GW2 arriving first at IN approaches 50% as expected, where a session breaks after having transmitted only approximately 3 packets on average.

It could be possible to implement a simple heuristic rule to eliminate the detected route race condition:

When a source node sends an RREQ to re-establish an external route, only the NAT-based gateways that have established NAT-state for that node are allowed to respond with a Proxy RREP.

Such a scheme could be implemented as follows: When a SN wants external access, it sends a RREQ with a “NAT-initiate” bit set, indicating that it has not yet established any NAT-state at any gateway. All NAT-based gateways reply with Proxy RREPs, and the first RREP returned from a gateway closest to the SN will form the outgoing route. When the SN sends out the first packet, NAT-state will be established at the gateway through which the packet is routed. If the route eventually times out, the SN sets a “NAT-reestablish” bit in the

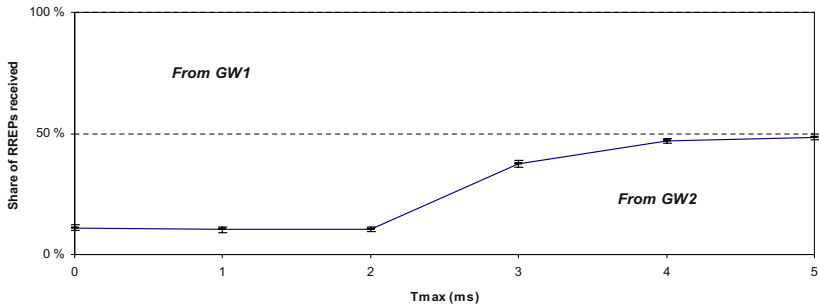


Fig. 3. The share of Proxy RREPs reaching the SN in Figure 1 (99% confidence interval).

RREQ in order to reestablish the route. Only the gateway that has established NAT-state for that SN is allowed to answer with a Proxy RREP.

Another way to eliminate the route race condition is to prohibit the use of Proxy RREPs. Instead, all NAT-based gateways would answer with RREPs establishing an outgoing route to the IP-address of the gateway. The SN can now tunnel the outgoing packets to the gateway, using for example IP-in-IP encapsulation [10] (Figure 4). The IP-address of the interface on the MANET-side of the NAT-based gateway should be used as destination IP-address of the encapsulating (outer) IP-header. The encapsulated (inner) IP-header is destined for the IP-address of XH and uses SN's IP-address as source address. The gateway must decapsulate the packets from SN *before* sending the inner IP-packets to the NAT-module.

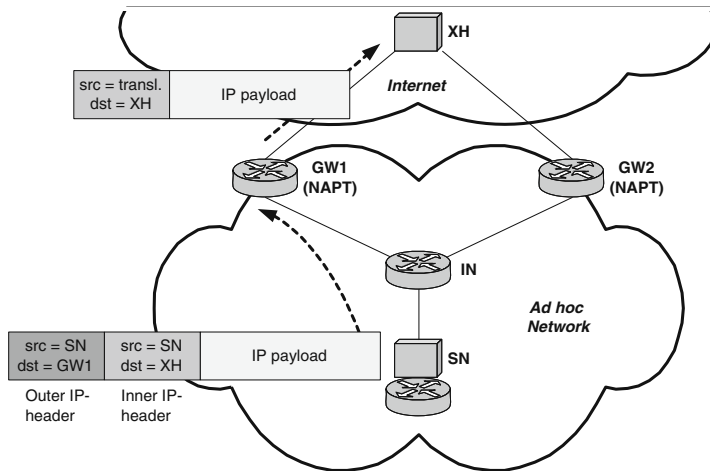


Fig. 4. Tunneling of packets to external host via gateway.

In Sub-section 3.2, it is discussed whether the Proxy RREP or the tunnelling approach is preferable.

Both these solutions allow the SN to be aware of the presence of gateways. Hence, instead of letting the gateways or the routing protocol determine whether a node is present on the MANET or not, it is possible to introduce an alternative 2-step SN-aware approach:

1. A source node first floods the MANET with RREQs to find a route to a node that is present. Gateways are not allowed to answer on behalf of external nodes.
2. If the source node does not succeed to locate the MANET locally, it issues a new RREQ with a “gateway”-bit set. When the bit is set, gateways are allowed to answer with RREPs on behalf of external nodes.

This approach allows source nodes to have better control with whether the packets are routed out of the MANET or not, which might also be a useful feature for name resolution in on-demand MANETs [11].

This solution also allows the SN to implement an integrated 1-step solution, i.e. the SN sets the “Gateway”-bit in every RREQ that it floods, and prioritizes an RREP returned directly from a targeted node that is present on the MANET.

3.2 NAT-Based Multi-homing with Two Competing Source Nodes

A site multi-homing test was performed, where the two source nodes (SN1 and SN2) both communicated with the same external host (XH). Two gateways (GW1 and GW2) were still present and both were reachable through the same Intermediate Node (IN). This is illustrated in Figure 5.

We first tested the heuristic rule using the Proxy RREP solution and no tunneling. This configuration experienced a problem caused by the fact that the IN can only have one active unicast route to the IP address of XH. This is a general limitation of unicast routing based entirely on destination IP-addresses. Due to the route race condition presented in the previous sub-section, SN1 may easily establish NAT-state over GW1 and SN2 over GW2. In this situation, SN1 and SN2 cannot communicate with XH simultaneously. The IN will either establish the outgoing route over GW1 in which the communication session of SN2 will break or over GW2 in which the communication session of SN1 breaks.

We did not experience such problems with the tunneling solution proposed in Sub-section 3.1. By tunneling packets out via the gateway, the IP-address of the XH will not appear in the routing protocol internally on the MANET. The routing protocol of the MANET will not be polluted with IP-addresses not really belonging to the MANET.

When we first introduced tunneling of outgoing packets we did not make any changes to the processing of incoming packets returned from the Internet. The NAT-based gateway would translate the incoming packets and inject them into the MANET. However, AODV uses the source and destination IP addresses of regular data-packets to reset the timers of the forwarding and return routes.

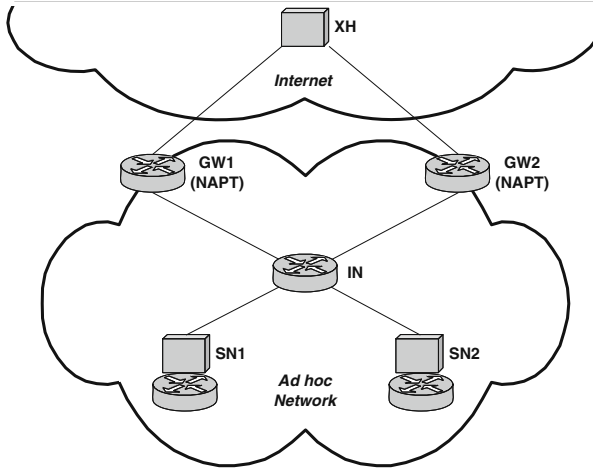


Fig. 5. A similar test-bed implementation as in Figure 1. Here there are two Source Nodes (SN1 and SN2) communicating with the same external host (XH) over the Internet.

Since an incoming packet destined for the SN carries XH's IP-address as the source address, the intermediate nodes were not able to update the timers of the outgoing route (since this is now a route to the IP-address of the NAT-based gateway).

The easiest way to deal with this problem is to let the NAT-based gateway tunnel the incoming packets to the destined source nodes, as well. Then, the gateway's IP-address appears as the source IP-address of the packets forwarded to the SN.

4 Summary and Proposed Solution

Our experiments show that the Proxy RREP solution without tunneling is not appropriate for routing between the source node and the gateways used for communicating with external hosts on the Internet. Instead a tunneled solution is required. The source node should tunnel traffic destined for the Internet to a selected gateway, and the gateway should tunnel return traffic from the Internet to the source nodes.

The following steps should be implemented to provide Internet access to MANETs:

- A SN should use the aforementioned 2-step solution when it searches for a destination. For detection of gateways the SN set a “Gateway”-bit in the RREQ when it believes that the targeted IP-address might be present on the Internet.

- Upon reception of a RREQ with this bit set, a gateway will return an RREP that establishes a route to its own IP-address. There will be no route race conditions between different gateways, which all have different IP addresses. Furthermore, there will not be race conditions between an RREP from a targeted node present on the MANET and an RREP from a gateway, independent of how the destination sequence numbers are set in the RREPs. The reason is that both the targeted node and the gateway will return an RREP that establishes a route to its own IP-address.
- A RREP returned from a gateway should contain an extension containing information about the capabilities of the gateway, e.g. whether it is a NAT-based gateway or a MIP-FA-based gateway. The source node might use this information when selecting an appropriate gateway that best matches its preferences. The extension might also include the IP-address that the source node searched for in the RREQ, so that the source node will be able to match a received RREP with a previously sent RREQ.
- To eliminate the need for additional signaling between SN and the NAT-based gateway, the gateway must set up a tunneling interface for the SN's IP-address as it returns an RREP to the SN. The tunneling interface enables the gateway to receive tunneled packets from the source node. Furthermore, the NAT will use it to tunnel packets returned from the Internet to the source node.
- The first packet that the SN to an external host establishes state in the NAT-module of the gateway after having been decapsulated.

A disadvantage is that tunneling comes at a cost, since every packet sent out of or into the MANET will carry an overhead of 20 bytes if IP-in-IP encapsulation is being used. Furthermore, the gateway will have to set up a tunneling-interface for each SN that wants to use it for Internet access, which will require it to store state information for every external connection. However, this can be managed by letting the gateway establish tunneling interfaces as soft state and remove interfaces that are not used within a certain time by setting an appropriate time out value.

The tunneling functionality can easily be integrated into the routing modules of MANET nodes. The fact that the tunneling solution is SN-aware (unlike a traditional NAT-solution) does not introduce any problem since it does not affect protocols and applications above the IP networking layer.

An IETF Internet Draft that describes explicit gateway discovery and tunneling in more detail has been published as a result our work. [12].

5 Applicability to MIP-FA-Based Multi-homing

Belding-Royer *et. al.* [2] do not explicitly consider multi-homing, and we therefore anticipate problems with race conditions also for this scheme. If there are several MIP-FA-based gateways at the same number of hops away from the SN, the outgoing route will be determined by a race condition and it might also alternate between different gateways. Since Mobile IP does not depend on outgoing

traffic to pass by the FA to which the mobile node is registered, this might not cause any significant problems (unless the outgoing traffic is subject to ingress filtering on the external network).

In a multi-homed scenario one must assume that NAT-based and MIP-FA-based gateways might be present simultaneously on a MANET (Figure 6). In this case, however, the race conditions may easily cause problems. If the SN uses NAT for external communication, outgoing packets escaping through the MIP-FA gateway will not be correctly translated. Similarly, if the SN uses the MIP-FA gateway for external communication, outgoing packets escaping through the NAT will be translated and will therefore not be recognized by the external host. However, by using explicit tunneling to the NAT-based gateway, the latter problem is eliminated.

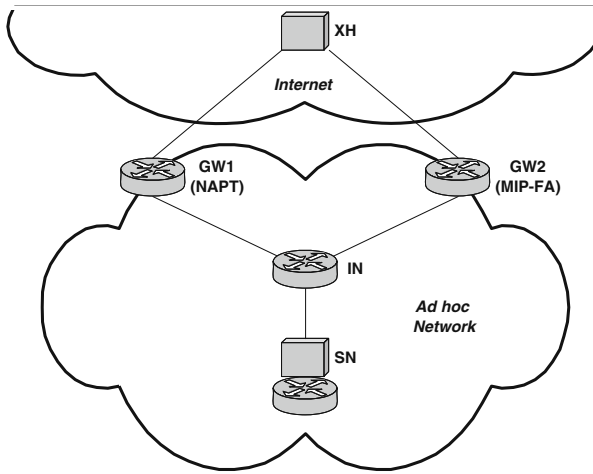


Fig. 6. A similar test set-up as in Figure 1. Here there are a NAT-based and a MIP-FA based gateways present on the MANET. The SN is capable of using both gateways. (The MIPv4 home agent of the SN on the Internet is not shown.)

To eliminate the problem of packets intended to pass through the NAT-gateway escaping through the MIP-FA gateway, we propose the following heuristic rule for MIP-FA gateways:

A gateway must not answer to an RREQ with a Proxy RREP unless the node originating the RREQ is registered with the MIP-FA on the gateway.

This rule also eliminates the problem of outgoing routes fluctuating between different MIP-FA-based gateways. Instead the outgoing traffic is forced to pass consistently out through the MIP-FA to which the SN is registered.

More work is required to determine whether this heuristic rule is sufficient for the MIP-FA-based solution, or whether such solutions also should be based on tunneling (as is proposed by the MIPMANET effort [3]). A source node might

for example register with two different MIP-FA-based gateways simultaneously (by setting the 'S'-bit of Mobile IPv4) in order to do a “make-before-break” change of gateways. In this situation, the race conditions between the different MIP-FA-based gateways will probably reoccur despite our proposed measures. Hence, tunneling might be required not only for NAT-based gateways, but also for MIP-FA-based solutions.

6 Applicability to DSR

Although our lab trials were based on AODV, the same arguments apply to DSR. However, the basic functionality that is part of the DSR routing protocol will inhibit that the aforementioned race conditions with multi-homing will occur.

A source node will use source routing - which can be considered as a sort of tunneling - to get the packet out through the selected gateway. The gateway will use source routing to get incoming packets forwarded to the source node.

The reserved 'L'-bit of DSR can be used to distinguish RREPs from gateways from a RREP returned directly from a node present on the MANET. However, an additional option containing the capabilities of the gateway might be required to make it easier for source nodes to select the right gateway with the right capabilities (e.g. describing to which extent the gateway supports MIP-FA, NAT or other kinds of gateway technologies).

7 Concluding Remarks

A number of proposed solutions for providing external connectivity to AODV-based on-demand networks allow a source node to send packets to an external host in the same way as to MANET-local hosts (i.e. without tunneling). By using Proxy RREPs, gateways respond to RREPs on behalf of external nodes, and all traffic destined for that node will be received by the gateway.

By simple replicable lab experiments with multi-homing, however, we have shown that this easily leads to routing race conditions. This again makes it difficult for a source node to control which gateway that will be used for external communication. Traffic might be directed over different gateways in a non-deterministic way.

With a NAT-based gateway, all outgoing packets belonging to the same communication session must pass through the same gateway, and the aforementioned race conditions must be avoided. The most obvious solution is to mandate that a source node tunnel packets to a selected gateway. This means that the source node must explicitly discover available gateways, and select one based on their different capabilities.

With a MIP-FA-based solution, the race conditions are less critical, since packets are allowed out through any gateway (unless the external network is subject to ingress filtering). We have proposed a simple heuristic rule to avoid routing race conditions, even when the Proxy RREP solution without tunneling

is being used. However, whether these measures are sufficient, or whether MIP-FA based solutions also should be based on tunneling, is a subject for further work.

Since test-beds are prone to non-deterministic changes of a number of physical parameter that are difficult to control, the analyses and results presented in this paper should be confirmed by simulations using an appropriate simulation tool. Simulations can provide numerical results that describe the multi-homing problems experienced in a test-bed in greater detail.

References

1. MANET Working Group of the *Internet Engineering Task Force (IETF)*, homepage, <http://www.ietf.org/html.charters/manet-charter.html>.
2. Belding-Royer *et al.*, "Global connectivity for IPv4 Mobile Ad Hoc Networks", *IETF Internet Draft*, draft-royer-manet-globalv4-00.txt, November 2001 (Work in Progress).
3. Alriksson, F., and Jönsson, U., "MIPMANET - Mobile IP for Mobile Ad Hoc Networks", Master of Science Thesis, Royal Institute of Technology (KTH), http://www.e.kth.se/~e94_fal/mipmanet.pdf, August 1999.
4. Perkins C. (ed.), "IP Mobility Support for IPv4", *RFC 3344*, *Internet Engineering Task Force (IETF)*, August 2002.
5. Srisuresh, P. and Holdrege, M., "IP Network Address Translator (NAT) Terminology and Considerations", *RFC 2663*, *Internet Engineering Task Force (IETF)*, August 1999.
6. Uppsala University's implementation (version 6) of AODV.
<http://user.it.uu.se/~henrikl/aodv/>
7. Perkins, C.E., Royer, E.M., and Das, S.R., "Ad-hoc On Demand Distance Vector (AODV) Routing", *RFC 3561*, *Internet Engineering Task Force (IETF)*, July 2003.
8. Johnson, D.B., Maltz, D.A., Hu, Y.-C. and Jetcheva, J.G., "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", *IETF Internet draft*. draft-ietf-manet-dsr-09.txt, April 2003 (Work in Progress).
9. Postel, J. and Reynolds, J., "Telnet Protocol Specification", *RFC 854*, *Internet Engineering Task Force (IETF)*, May 1983.
10. Perkins, C.E., "IP Encapsulation within IP", *RFC 2003*, *Internet Engineering Task Force (IETF)*, October 1996.
11. Engelstad, P.E. and Egeland, G., "Name Resolution in On Demand MANETs", *IETF Internet draft*, draft-engelstad-manet-name-resolution-00.txt, February 2003 (Work in Progress).
12. Engelstad, P.E. and Egeland, G., "Explicit Gateway Discovery in IPv4 On Demand MANETs", *IETF Internet draft*, draft-engelstad-manet-gateway-discovery-v4-00.txt, October 2003 (Work in Progress).

Efficient Management of Domain Foreign Agents in Mobile Computing Environment Using Load Balance

Yong Chul Kim¹, Min Gyo Chung², and Jun Hwang²

¹ Dept. of Computer Science, ChungAng University, Seoul, Korea
pertzs@sslabs.cse.cau.ac.kr

² Dept. of Computer Science, Seoul Women's University, Seoul, Korea
{mchung, hjun}@swu.ac.kr

Abstract. Mobile IP is a protocol standard designed to be used in a mobile computing environment. However, Mobile IP has a drawback to incur a lot of handoff delays and waste network resources, since CoA registration packets need to go through a HA first whenever a mobile node moves. To solve such problem, this paper proposes a new scheme that, for intra-domain movement, efficiently performs local handoff without notifying the HA. Specifically, based on the notion of load balance, the proposed scheme allows every FA in a domain to become the root FA(a.k.a. Domain FA) dynamically, thus distributing the registration task into many other foreign agents. Our simulation results show that the proposed method proves to reduce registration packets by approximately 7–15% more than existing methods.

1 Introduction

Mobile IP is a standard designed to transparently provide the mobility for the wireless devices on the Internet. One of the drawbacks of Mobile IP is that whenever a mobile node moves to a foreign network, it should register its new CoA(Care-of Address) with its home agent [4]. During frequent movement of the mobile node, if the home network is far away from the foreign network, this address registration can incur a few problems such as exorbitant handoff delays, packet loss, etc. To alleviate these long-standing problems involved in CoA registration process [2,6], this paper proposes a new scheme that, for intra-domain movement, efficiently performs local handoff without notifying the home agent. Specifically, based on the notion of load balance, the proposed scheme allows every foreign agent in a domain to become the domain root dynamically, thus distributing the registration task into many other foreign agents. The dynamic root assignment through load balancing ultimately leads to fast network response due to less frequent transmission of registration packets.

This paper is organized as follows. Section 2 briefly describes Mobile IP and some approaches to solve CoA registration problems. Section 3 gives a detailed description of a new scheme proposed in this paper. Extensive experimental results and analyses are presented in Section 4. Lastly, Section 5 summarizes this paper.

2 Relevant Researches

2.1 Mobile IP

Mobile IP is designed for a mobile node to have two IP addresses: home address and care-of address(CoA). The home address is an IP address to uniquely identify the mobile node, while the care-of address is an IP address used to forward packets delivered with home address to the mobile node when the node is not attached to its home network[1]. Mobile IP requires two agents for a mobile node: home agent(HA) and foreign agent(FA) [4, 5, 6]. The home agent is responsible for the management of the home address and the CoA of the mobile node. The foreign agent is a mobility agent on the foreign network to which the mobile node is currently attached, and offers its IP address to the mobile node as the CoA. The home agent can receive all the packets destined to the mobile node and if needed, can redirect them to the foreign agent, which then forwards the received packets to the final mobile node.

One shortcoming in Mobile IP, also known as *triangle routing*, is that the packets destined to the mobile node must be routed through the home agent, even when the mobile node and the correspondent node are physically near to each other. The triangle routing problem brings about a lot of communication delays [3,4]. To avoid this triangle routing, a mechanism called *route optimization* was suggested. However, the route optimization scheme still has some problems with micro mobility, smooth handoff, and security.

2.2 Hierarchical Mobility Management

There are a lot of overheads (e.g., protocol latency) involved in registering a new CoA after the detection of a mobile node's movement. In particular, if the foreign network is far away from the home network or if the handoff is repeatedly performed due to small many movements, the overheads will be enormously significant. In this context, *hierarchical mobility management* was devised to reduce such overheads by registering the new CoA with a specific local agent on the foreign network, not with the home agent on the home network.

The hierarchical mobility management scheme is based on the hierarchy of foreign agents in a domain(Figure 1)[1,3,4,7]. The root of the tree, which is called root FA or DFA(Domain Foreign Agent), is responsible for all foreign mobile hosts within the domain. The first time when a mobile node moves into a foreign domain, a new CoA of the mobile node is registered with the home agent. Hereafter, for every intra-domain movement, the corresponding CoA is registered with the DFA, not the home agent. In this way, the hierarchical scheme hides the mobility within the foreign domain from the home agent, and thereby leads to the dramatic reduction of CoA registration-related time and messages. However, it has serious drawbacks that the DFA is only a single node to be responsible for the address registration of all mobile nodes in the domain, and therefore, the DFA is easily overloaded with the heavy traffic from the correspondent nodes.

3 New Scheme for Management of Domain Foreign Agents

Figure 1(a) illustrates a scenario in which a mobile node performs a handoff from one foreign agent labeled FA8 to another foreign agent labeled FA9, using a conventional mobility management mechanism. The request message for address registration is generated by FA8 and passes up through FA4, FA2, and FA1 to the home agent, HA. In response to the request, the home agent sends an acknowledgement, which will come down through FA1, FA2, and FA4 to the destination agent, FA9. The conventional method forces all messages to pass through the root FA, FA1, which will soon be a bottleneck for message traffic. Additionally, the above scenario requires as many as 8 message transmissions, though the FAs participated in the handoff are adjacent to each other.

For the purpose of the new scheme presented in this paper, assume that all foreign agents are structured in a form of binary tree, and that each node in the tree is labeled in the order as in Figure 1. Then, the basic idea of the new scheme is that based on the notion of load balance, it allows every FA in a domain to become the DFA dynamically, thus distributing the CoA registration task into many other foreign agents in a domain.

Consider Figure 1(b) that depicts how the proposed scheme works when the same scenario as in Figure 1(a) is applied. The proposed scheme first figures out the common ancestor FA, FA4, between FA8 and FA9, then designates the ancestor FA as a new DFA, and next lets the new DFA forward the registration message to the home agent. The registration message will include all information specific to this handoff, such as source FA, destination FA, new DFA, etc. In return for the registration request, the home agent will communicate with new DFA (i.e., FA4) for the delivery of an acknowledgement. Subsequently, the new DFA forwards the received acknowledgement to the destination FA, FA9.

Note some benefits from using the proposed method. The proposed method can enable the traffic having passed through one bottlenecked DFA to be now distributed into each and every FA in a domain, as well as can alleviate some stubborn problems in mobile environment such as packet loss and delay, thanks to fewer number of message transmission and faster response time.

One core part of the proposed scheme is the algorithm to select a new DFA. Basically, the algorithm finds the greatest common FA for the given two input FAs. The detailed steps and the pseudo code for implementing the steps are described below.

- Assume two input nodes are FA_p and FA_f , where FA_p represents the FA at which a mobile node is now present, and FA_f represents the FA to which the mobile node is going to move.
- Make FA_p and FA_f be positioned at the same tree level by moving only the higher level FA up to the lower level along the tree branches (recall that the level of the root is lowest). The modified new FAs are then denoted again as FA_p^* and FA_f^* .
- Find out the nearest parent, FA_{root} , of two modified FAs (i.e., FA_p^* and FA_f^*).
- Choose FA_{root} as a new DFA.

```
Find-Nearest-Parent-Node( ) {
    divide i, j by 2 until i = j;
```



```

    set node i or j as as a parent;
}
...
if(Level(i) = Level(j))
    Find-Nearest-Parent-Node( );
else if(Level(i) < Level(j)) {
    do j = ⌈j/2⌉ until (Level(i) = Level(j));
    Find-Nearest-Parent-Node( );
} else {
    do i = ⌈i/2⌉ until (Level(i) = Level(j));
    Find-Nearest-Parent-Node( );
}

```

Consider Figure 1(b) again to see how the DFA selection algorithm works when a mobile node moves from FA4 to FA15. According to the previous notation, the algorithm starts with $FA_p = FA4$ and $FA_f = FA15$, then finds out $FA_p^* = FA4$ and $FA_f^* = FA7$ as two modified FAs at the same tree level. At the last step, the algorithm yields $FA_{root} = FA1$ as the closest parent of FA_p^* and FA_f^* , and declare that $FA_{root} = FA1$ now becomes a new DFA.

The above scenario (switching FAs from FA4 to FA15) gives a kind of worst-case examples. However, in the real world mobile computing environment, it is believed that the circumstances like this scenario rarely happen because the mobile node mostly roams around adjacent foreign agents.

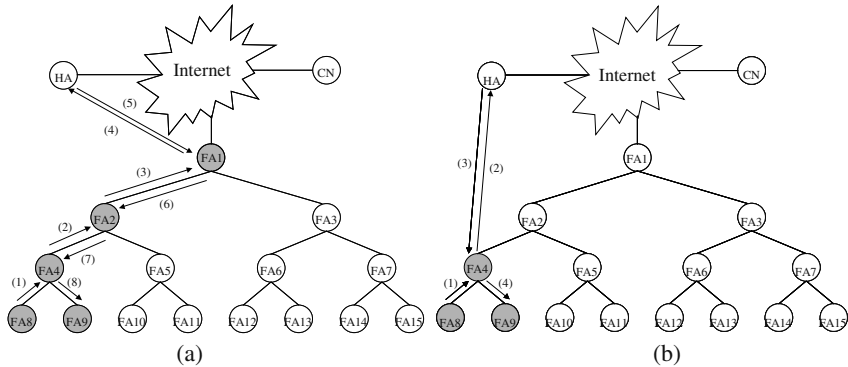


Fig. 1. (a) A scenario for registering CoA in a tree-structured domain by existing methods. (b) An exemplar scenario for registering CoA in a tree-structured domain by the new method

4 Simulation Results

This section presents some experimental results comparing the proposed method and existing methods. Figure 2 provides the result for the best scenario in which the handoff occurs between an FA and its child FA. Regarding existing methods, as the tree level of handoff-related FA nodes gets deeper and deeper, the number of mes-

sages transmitted increases linearly because the messages must climb up the branches of the tree to the root node. On the contrary, the proposed method produces a constant number of messages because the DFA is not fixed to the root node, but can become any node in the tree according to the DFA selection algorithm presented in this paper. Theoretically, the above scenario is plausible, but in most practical cases, the message count of the proposed method is expected to come between the two lines in Figure 2.

Meanwhile, Figure 3 plots the performance comparison between the existing methods and the proposed method when the two FAs around which the mobile node will roam are selected at random. The horizontal axis represents the number of FA nodes included in a domain and the vertical axis represents the ratio of message volumes created by the new method to the message volumes created by the old method, with the rate of 1.0 fixed for the old method. For each group of FA nodes on the horizontal axis, we made a random selection of FA_p and FA_r nodes from the group 10000 times, and applied the proposed scheme to each selected pair of FA nodes. The sum of the message volumes generated by each pair is computed and then divided by 10000 to get an average message volume. The ratio of two average message volumes from old methods and the new method is finally computed and plotted into the graph. Simulation results tell us that the ratios on the vertical axis corresponding to the newly proposed method fast converge into equilibrium as the number of nodes increases. Specifically, when the node count goes beyond 32768, the ratio approaches to a constant, 0.93, which proves that the proposed method can reduce the total message traffic by at least 7%.

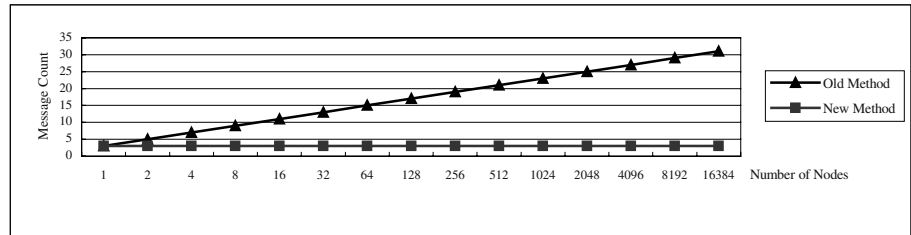


Fig. 2. Comparison of existing methods and the new method: the best scenario

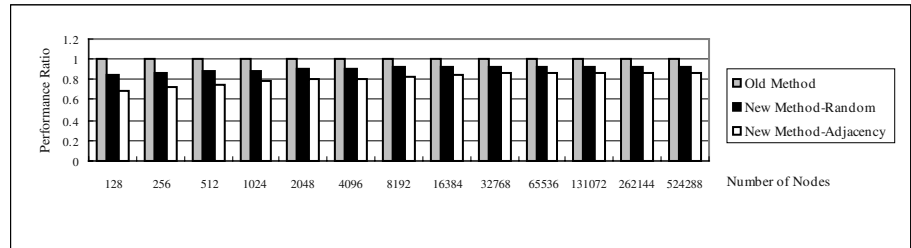


Fig. 3. Comparison of existing methods and the new method: foreign agents participated in the handoff are selected at random or adjacent to each other

The improvement of 7% is made when the handoff happens between two arbitrary FAs in a random fashion. However, it is easily observed in the real world that the handoff is often performed between adjacent FAs. Figure 3 also shows the performance comparison when the adjacent handoff is executed. We can infer similar interpretations as the previous random case. That is, when the node count is greater than 32768, the ratio fast converges to a constant, 0.85, which corresponds to approximately 15% reduction in total message traffic by the proposed method.

5 Conclusion

To meet the ever-increasing demand for mobile computing, Mobile IP was introduced and is now gaining popularity more and more. However, Mobile IP has some shortcomings related to the mobility management.

This paper proposes a new mobility management scheme that, for intra-domain movement, efficiently performs local handoff without notifying the home agent. Specifically, based on the notion of load balance, the proposed scheme allows every foreign agent in a domain to become the domain root dynamically, thus distributing the registration task into many other foreign agents.

Our simulation results exhibit that our proposed method proves to reduce the total message traffic by approximately 7-15% more than existing methods. The proposed method prevents the message traffic from swamping the root node in the tree. Rather, the traffic is now fairly distributed into each FA in a domain. Plus, the proposed method can alleviate some stubborn problems in mobile environment such as packet loss and delay, thanks to less number of message transmission.

Acknowledgement. This work has been supported in part by 2003 Special Research Program of Seoul Women's University.

References

1. C. Perkins, "Mobile IP," IEEE Communications Magazine, 84-99, May 1977
2. C. Perkins, "Mobile Networking Trough Mobile IP," IEEE Internet Computing, 58-68, 1998
3. C. Perkins, "Mobile IP - Update," IEEE Communications Magazine, 66-82, May 2002
4. C. Perkins, "Optimized Smoothed Handoff in Mobile IP," Proc. Of IEEE International Symposium on Computers and Communications, 340-346, 1999
5. C. Perkins, "IP Mobility Support," RFC 2002, October 1996
6. C. Perkins, "IP Mobility Support for IPv4," RFC 3220, January 2002
7. Eva Gustafsson, A. Jonsson, and C. Perkins, "Mobile IPv4 Regional Registration," IETF Internet Draft, 2001

Node Synchronization Based Redundant Routing for Mobile Ad-Hoc Networks*

Wonjong Noh, Yunkuk Kim, and Sunshin An

Computer Network Lab, Dep. Of Electronics Engineering, Korea University
Sungbuk-gu, Anam-dong 5ga 1, SEOUL, KOREA, Post Code: 136-701,
Phone: +82-2-925-5377, FAX: +82-2-3290-3674
{nwj, dbs1225, sunshin}@dsys.korea.ac.kr

Abstract. This paper proposes a new on-demand redundant-path routing protocol considering node synchronization based path redundancy as one of route selection criteria. Path redundancy implies how many possible redundant paths may exist on a route to be built up. Our proposal aims to establish a route that contains more redundant paths toward the destination by involving intermediate nodes with relatively mode adjacent nodes in a possible route. Our approach can localize the effects of route failures, and reduce control traffic overhead and route reconfiguration time by enhancing the reachability to the destination node without source-initiated route re-discoveries at route failures. We have evaluated the performance of our routing scheme through a series of simulations using the Network Simulator 2 (ns-2).

1 Introduction

A mobile ad hoc network [1] is a collection of wireless mobile nodes forming a temporary network without the aid of any established infrastructure. Adjacent nodes communicate directly between one another over wireless channels. However, since the transmission range of nodes is limited, the nodes that are not neighboring need routing supported by intermediate nodes for communications. Due to the mobility of nodes, the topology of connections between communicating nodes may be quite dynamic. In a dynamic ad hoc network, route re-discoveries due to route failures may incur heavy control traffic through the network and cause the increase of packet transmission delay. Hence, it is quite required to reduce the number of route re-discoveries by maintaining multiple redundant paths, establishing alternate route promptly and localizing the effect of the failures. Numerous ad hoc routing protocols have been developed [2, 3, 4, 5, 6, 7] but most of them don't deal with route reconfiguration by dynamic topology.

This paper presents a new on-demand redundant-path ad-hoc network routing protocol that provides dynamic and fast route reconfiguration using information about

* An unabridged version of this paper is available at the following URL:
<http://nlab.korea.ac.kr/~angus/wons2003-full.pdf>

redundant paths maintained at a source and intermediate nodes on initial route. Our protocol can establish a route consisting of intermediate nodes with relatively more node-synchronized neighboring nodes and hence it provides more redundant paths toward the destination. We introduce a new routing metric, ‘path redundancy’, which implies how many possible redundant paths on a route to be built up. The remainder of this paper is organized as follows. The following section describes node synchronization concepts. Section 3 presents our routing procedures. Finally we present the simulation result and conclusions.

2 Node Synchronizations

2.1 Definitions

\vec{v}_i , \vec{v}_{ij} : Node(i)’s mobility vector, node(i)’s relative mobility vector with respect to node (j)

$T_{data,i}^e$: Expected whole data transmission time at node(i)

$T_{power,j}^e = \int_0^\infty P(T_{power,j} > t \mid T = T_c) dt$: Expected power alive time at node (j)

$T_{link,i,j}^e = \int_0^\infty P(T_{link,i,j} > t \mid T = T_c) dt = \int_0^\infty P\left(\min\left\{T \mid \left|\vec{d}_{ij}(T_c) + \int_{T_c}^{T_c+T} \vec{v}_{ij}(t) dt\right| \geq r_i\right\} > t\right) dt$

: Expected link-alive time between neighboring two nodes, node(i) and node(j),

where, $\vec{d}_{ij}(T_c)$ is node(j)’s location vector with regard to node(i) at current time T_c .

r_i is transmission range of node(i)

$R_{comb,i}^e$: Expected combinatorially stable region

$R_{\theta,i,j}^e$: Estimated maximal pie-type sub-region inside node(i)’s neighborhood region

2.2 Synchronizations

When $T_{data,i}^e$ is less than or equal to $\min\{T_{link,i,j}^e, T_{power,j}^e\}$, node(j) is considered to be time synchronized to node(i). Spatial synchronization is related to each node’s

current position. When the node(j) is located in intersection area between combinatorially stable region [8] and estimated maximal pie-type sub-region inside neighborhood region, $R_{comb,i}^e \cap R_{\theta,i,j}^e$, it is considered to be spatially synchronized to node(i). Mobility synchronization is related to each node's mobility vectors. If the inequality $\left| \arccos\left(\frac{\vec{v}_i \cdot \vec{v}_j}{|\vec{v}_i| \cdot |\vec{v}_j|}\right) \right| \leq \frac{\pi}{2}$ is satisfied, node(i) and node(j) is mobility synchronized.

2.3 Path Redundancy

$$\sum_{i \in \text{path}(p)} \sum_{j \in \text{nbd}(i)} \text{sync}(i, j) \quad (1)$$

$$\text{sync}(i, j) = \alpha(i, j) \cdot I_{\text{time_sync},i,j} + \beta(i, j) \cdot I_{\text{spatial_sync},i,j} + \gamma(i, j) \cdot I_{\text{mobility_sync},i,j} \quad (2)$$

$$\arg \max_{\text{path}(p) \in \text{paths}(src, dst)} \left\{ \sum_{i \in \text{path}(p)} \min \left\{ \sum_{j \in \text{nbd}(i)} \text{sync}(i, j), \text{UpperLimit} \right\} \right\} \quad (3)$$

Formulation(1) denotes the path(p)'s redundancy degree. Where, function I is a synchronization indicator function. If node(i) and node(j) is synchronized with respect to time, spatial and mobility, $I_{\{\text{time,spatial,mobility}\}\text{-sync},i,j}$ is 1, otherwise 0. $\alpha(i, j)$, $\beta(i, j)$ and $\gamma(i, j)$ are weigh functions with respect to time, spatial and mobility synchronization. Formulation(3) shows how to choose an optimal route from source node to destination node. To prohibit the case of a path's redundancy degree is severely influenced by a specific node having especially large node redundancy degree, we use the upper limit on each node's redundancy degree.

3 Node Synchronization Based Redundant Routing

3.1 Route Request Process

A node initiates route establishment procedure by broadcasting a route setup message, Route Setup (RS) packet. An RS packet is flooded throughout the network as shown in Fig.1 and carries the information about redundancy degree and hop distance of nodes that it goes through. Any node that receives an RS packet does the following:
Case 1: If the node recognizes its own address as the intermediate node address, the node records the address of the neighbor node from which it received the RS packet as the upstream node. The recorded node address will be used to build a route during

the route reply process. Then, it adds its own redundancy degree to that of the RS packet and broadcasts the updated packet to its neighbor nodes.

Case 2: If the node has already received the RS packet with the same identification, it records the address of the node from which it received the packet as a redundant upstream node and then drops that packet. The recorded node address will be used to build a redundant path if this node is involved in the selected route.

Case 3: If the node recognizes its own address as the destination address, it records the forwarding node address, hop count and path redundancy of the packet.

For the purpose of optimal route selection, the destination will wait for a certain number of RS packets to reach it after receiving the first RS packet. The destination node can receive several RS packets transmitted along different paths from the source node. An RS packet delivered along the shortest route will early reach the destination node and RS packets representing routes with more redundant links may come to later. The destination node adopts the RS packet that reached it later, but contained larger path redundancy per hop, and sends a Route Reply (RR) packet back to the source node via the node from which it received the RS packet.

3.2 Route Reply Process

A route containing redundant paths toward the destination is established during the route reply process. After selecting the optimal path using equation(3), the destination node initiates the route reply process by sending an RR packet back to the source node via the node from which it received the corresponding RS packet. An RR packet is forwarded back along the transit nodes the RS packet was traversed. An RR packet carries the hop distance from the destination to the node that received the RR packet. The hop distance is incremented by one whenever the RR packet is forwarded at each intermediate node. Any node that receives an RR packet does the following.

Case 1: If the node recognizes itself as the target node of the received RR packet, it records the forwarding node address of the packet as the next hop for the destination. Then, the node increments the hop distance of the received packet and sends the updated packet to its upstream node, which was recorded during the route setup process. Moreover, if the node has any redundant upstream node recorded, it generates and sends the Redundant Route Reply (RRR) packet to the redundant neighbor node(s). The hop distance of the RR packet is copied into the hop distance field of the RRR packet.

Case 2: If the node recognizes its own address as the source, it records the forwarding node address of the RR packet as the next hop for the destination in the route table.

Case 3: If the node is not targeted, the node discards the RR packet.

The RRR packet is used to setup a redundant path of a route. An RRR packet is originated from only nodes along a main route if they have redundant nodes in the upstream direction. RRR packets are forwarded at redundant nodes toward the source node. Any node that receives an RRR packet does the following:

Case 1: If the node exists along the main route, it creates the redundant route table (RRT) entry, which is a set of redundant neighbor nodes for the destination. A

redundant next hop field of the RRT entry is filled with the forwarding node address of the RRR packet.

Case 2: If the node is along a redundant path and has not received the RRR packet, it records the forwarding node address of the RRR packet as a redundant next hop for the destination in the RRT entry. Then, it forwards the packet to the upstream nodes.

Case 3: If the node is along a redundant path and has already received the RRR packet with the same identification, it discards the packet. This means that a redundant path cannot have any redundant path for itself.

Case 4: If the node is not targeted, it discards the RRR packet.

3.3 Route Reconfiguration

Failure notification is progressed when a failure-detecting node or a node that received a Failure Notification (FN) packet does not have any redundant path for the destination. Route failure information is carried using an FN packet and stored in the failure record. Route failure information is carried using a Failure Notification (FN) packet and stored in the failure record. Route failure information includes the information about a failure-detecting node, whether the failure-detecting node is along a main route or not, and intermediate transit nodes that an FN packet is propagated through. Any node that receives an FN packet does the following.

Case 1: If the node is along a main route (shortly, main node) and the FN packet originated from a main node, it records the failure information and finds an alternate redundant path.

Case 2: If the node is a main node and the FN packet originated from a node along a redundant path (shortly, redundant node), it removes the corresponding redundant path information from the RRT entry.

Case 3: If the node is a main node and the FN packet originated from a node along an active redundant path (shortly, active redundant node), it records the failure information and finds an alternate redundant path.

Case 4: If the node cannot find a redundant path, it adds its node address to the FN transit node list of the FN packet and propagates the updated packet. Moreover, it deletes all the information about the failed route.

Case 5: If the node is an active redundant node, it deletes the corresponding Routing Table (RT) entry and RRT entry and adds its node address to the FN transit node list of the FN packet and propagates the updated packet.

Case 6: If the node is an inactive redundant node, it deletes the corresponding RRT entry and broadcasts the packet.

4 Conclusion

This paper presented a new on-demand ad-hoc network routing protocol that can accomplish dynamic and fast route reconfiguration using information about redundant paths maintained at a source node and intermediate nodes on the main route. The

proposed routing protocol can establish a main route with relatively more synchronized redundant paths using a new routing metric of path redundancy. Our approach can localize the effects of route failures and cut down route reconfiguration time by raising the reachability to destination nodes without route re-discoveries in the event of the route failures. We conducted a performance evaluation of our protocol through a simulation using NS-2 [9]. We measured performance comparing with DSDV, AODV, SMR and TORA. The simulation results say that the use of redundant routing considering node synchronization can reduce control traffic overhead and enhance packet delivery ratio and end-to-end delay in mobile ad-hoc networks. The details of simulation results are available at <http://nlab.korea.ac.kr/~angus/wons2003-full.pdf>

References

1. Introduction To Wireless and Mobile Systems, D. P. Agrawal, Qing-An Zeng, Brooks/Cole, a division of Thomson Learning, 2003.
2. J. Broch, D.B. Johnson, and D.A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," Internet Draft, draft-ietf-manet-dsr-00.txt, March 1998.
3. D. Jhonson and D. A. Maltz, Dynamic source routing in ad hoc wireless networks, in Mobile Computing, Kluwere Academic Publishers, 1996.
4. C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced vector routing (DSDV) for mobile computers," ACM SIGCOMM, Oct. 1994.
5. C. Perkins and E. Royer, "Ad Hoc On-Demand Distance Vector Routing," Proceeding of IEEE WMCSA, Feb. 1999.
6. E.M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Networks," IEEE Personal Communications, vol. 6, no. 2, pp. 46–55, April 1999.
7. C-K Toh, "Associativity-based routing for ad hoc mobile networks," Wireless Personal Communications, vol. 4, no. 2, pp. 1–36 Mar. 1997.
8. Satyabrata Chakrabarti, Amitabh Mishra, "QoS issues in ad-hoc wireless networks", IEEE communications magazine, Feb., 2001
9. The VINT Project, The UCB/LBNL/VINT Network Simulator-ns (version2), <http://www.isi.edu/nsnam/ns>.

A New Backoff Algorithm to Guarantee Quality of Service over IEEE 802.11 Wireless Local Area Networks

Kil-Woong Jang

Dept. of Applied Science, Korea Maritime University, Busan, Korea
jangkw@bada.hhu.ac.kr

Abstract. In this paper, we propose a new backoff algorithm to guarantee quality of service over IEEE 802.11 wireless local area networks. It is designed to carry out the proposed backoff algorithm, which changes the contention window size for backoff time using information of basic service area, such as number of stations. We evaluate the performance of the proposed algorithm using Markov model analysis and compare it with the IEEE 802.11e backoff algorithm. The numerical results show that the proposed algorithm is able to offer better performance than the conventional backoff algorithm in terms of the throughput.

1 Introduction

Wireless local area networks (WLANs) provide high bandwidth and real-time multimedia applications for users in a limited geographical area. They are supported by two standards: the IEEE 802.11 standard [4,5] and High Performance Radio LAN (HIPERLAN) Type 2 [6]. In particular, the IEEE 802.11 standard provides two service types: asynchronous and delay bound. The asynchronous service is provided by the distributed coordination function (DCF), which implements the basic access method of the IEEE 802.11 media access control (MAC) protocol. It is also known as the carrier sense multiple access with collision avoidance (CSMA/CA) protocol.

In CSMA/CA, each station seeking access to the medium selects a random time slot within the contention window (CW). The station that selects the shortest random time will gain access for transmission; the others stop their backoff times until the transmission is finished and wait for the remaining time in the following cycle. However, this contention-based MAC protocol is unable to guarantee quality of service (QoS) for time-sensitive traffic. In WLANs, all stations must compete for access to the shared medium. Therefore, the competition may cause longer transmission delay and lower throughput due to collisions.

In this paper, we propose a new backoff algorithm in order to improve the throughput of real-time traffic by resizing the CW . Specifically, we develop a Markov model of the IEEE 802.11 DCF protocol. We then use this model to analyze the properties of the proposed algorithm.

2 IEEE 802.11e Backoff Algorithm

The IEEE 802.11e standard defines a single coordination function, called the hybrid coordination function (HCF), for the QoS provisioning. The HCF is a function that combines aspects of the DCF and point coordination function (PCF) to provide the selective handling of frames required for the QoS facility. The HCF uses a contention-based channel access mechanism, referred to as the enhanced DCF (EDCF). A enhanced station (QSTA) operates according to the same general rules defined for EDCF by providing separate output queues. Each queue initiates a EDCF state machine that contends for the wireless medium with AIFS[i] rather than DIFS, where i is traffic priority and is determined by traffic category (TC) defined as the IEEE 802.11e standard. In addition, it employs a $CW_{min}[i]$ rather than a CW_{min} between queues within an QSTA.

The backoff algorithm shall be invoked when a transmitting station infers a failed transmission. To begin the backoff algorithm, the QSTA shall set its backoff timer to a random backoff time using the following equation:

$$T_{backoff}[i] = Random(i) \times SlotTime . \quad (1)$$

where $Random(i)$ is a pseudo random integer drawn from a uniform distribution over the interval $[0, CW[i]]$. $CW[i]$ is an integer within the range of values of the management information base (MIB) attributes a $CW_{min}[i]$ and a $CW_{max}[i]$.

To compute the new $CW[i]$ value, denoted $CW_{new}[i]$, from the old $CW[i]$ value, denoted $CW_{old}[i]$, in the event of a collision, a station shall choose a value of $CW_{new}[i]$ that meets the following criterion:

$$CW_{new}[i] = ((CW_{old}[i] + 1) \times PF) - 1 . \quad (2)$$

where the persistence factor, PF , is computed using the procedure described in [5]. The values of $CW_{min}[i]$, AIFS[i] and PF are transmitted by an enhanced access point (QAP) using the management frame with the QoS parameter set element.

3 Proposed Backoff Algorithm

In this section, we present the proposed backoff algorithm to increase the throughput for time-sensitive traffic over IEEE 802.11e WLANs. Our basic idea is to allow the backoff algorithm to resize the CW according to a number of stations.

Of IEEE 802.11e standard management frames, the management frame with the QBSS load element contains information on the current station population in the QoS basic service set (QBSS). The station count field in the frame indicates the total number of normal stations (STAs) and QSTAs currently associated with this QBSS. Using the station count field of this frame, the QAP can control the CW size under the certain status of transmission.

The proposed algorithm is attempted as follows. We first assume that the traffic is divided into two types: real-time and non-real-time traffic. According to

the traffic type, the proposed algorithm carries out a retransmission procedure using the different CW . We distinguish whether a station can support the QoS or not. If a station is a STA, the backoff procedure is conducted according to the conventional backoff procedure. The set of CW values shall be sequentially ascending integer powers of the value of default PF , minus 1, beginning with a CW_{min} value and continuing up to and including a CW_{max} value. In this paper we assume that the value of the default PF is 2.

If a station is a QSTA, the backoff procedure carries out the proposed algorithm. In the proposed algorithm, for real-time traffic, QAPs set a boundary, called a threshold, ϕ , to distinguish the CW size under two states of transmission in QBSS: the idle and busy states. If a number of stations, n is lower than ϕ , it is called an idle state. Otherwise, it is called a busy state.

In an idle state, to reduce the waiting delay during the backoff procedure, a high-priority station with real-time traffic has a smaller PF value than the default PF value. However, if traffic occurs heavily, collisions will be increased because the CW for real-time traffic has a low value. Therefore, the amount of the transmitted frame is reduced due to high collisions. To reduce the occurrence of collisions, in a busy state the CW value is increased as the QAP increases the PF value.

Proposed Backoff Algorithm()

```

1  if (IsQSTA() == True)    // if a station is QSTA
2      if (s(t) > 1)        // s(t): backoff retry count
3          if (s(t) <= m)    // m: max. of s(t)
4              if (CWold < CWmax) {
5                  if (TC == Real-time Traffic)
6                      if (n < Threshold)
7                          Set PF;          // PF < Default PF
8                      else Set PF;          // PF > Default PF
9                  else Set PF = Default PF;
10                 CWnew = ((CWold+1)*PF)-1; }
11             else CWnew = CWmax;          // if CWold >= CWmax
12             else Discard Packet;          // if over m
13         else CWnew = CWmin;              // if first trans.
14     else IEEE 802.11 backoff procedure;  // if not QSTA

```

4 Performance Evaluation

We present a Markov model to obtain the throughput, γ . A Markov model for backoff time was proposed in [1-3] to analyze the performance of the IEEE 802.11 protocol that only employs the DCF. In order to accurately analyze the performance of the DBA, we adopt the Markov model proposed in [1-3] and we present the modified Markov model to be suited the proposed algorithm, as shown in Fig. 1.

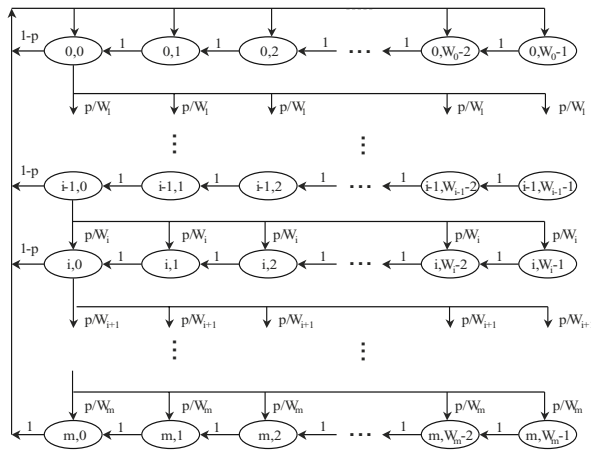


Fig. 1. Markov model for backoff time

We consider a fixed number n of contending stations. Let $s(t)$ be the stochastic process representing the backoff retry count of the station at time t and $c(t)$ be the stochastic process representing the backoff time counter for a station at time t . In addition, let m be the maximum value of the backoff retry count. We assume that each frame collides with constant and independent probability P_c . Once independence is assumed and P_c is supposed to be a constant value, the tuple $\{s(t), c(t)\}$ is a discrete-time Markov chain with transition probabilities [1]. In this Markov chain we set $P\{i_1, j_1 \mid i_0, j_0\} = P\{s(t+1)=i_1, c(t+1)=j_1 \mid s(t)=i_0, c(t)=j_0\}$. To describe the decrement of the backoff time counter, we have

$$\begin{cases} P\{i, j \mid i, j+1\} = 1 & (0 \leq i \leq m, 0 \leq j \leq W_i - 2) \\ P\{0, j \mid i, 0\} = 1 - P_c/W_0 & (0 \leq i \leq m-1, 0 \leq j \leq W_0 - 1) \\ P\{i, j \mid i-1, 0\} = P_c/W_i & (1 \leq i \leq m, 0 \leq j \leq W_i - 1) \\ P\{0, j \mid m, 0\} = 1/W_0 & (0 \leq j \leq W_m - 1) \end{cases} \quad (3)$$

Here, we let $c_{i,j}$ be the stationary distribution of the Markov chain and we have

$$c_{i,j} = \lim_{t \rightarrow \infty} P\{s(t) = i, c(t) = j\} \quad (1 \leq i \leq m, 0 \leq j \leq W_i - 1). \quad (4)$$

Due to the Markov chain regularities, we can obtain a solution for this Markov chain. The solutions are derived as follows:

$$c_{i,0} = P_c c_{i-1,0} \rightarrow c_{i,0} = P_c^i c_{0,0} \quad (1 \leq i \leq m). \quad (5)$$

$$c_{i,j} = \frac{W_i - j}{W_j} \begin{cases} (1 - P_c) \sum_{k=0}^{m-1} c_{k,0} + c_{m,0} & (i = 0) \\ P_c c_{i-1,0} & (0 < i \leq m) \end{cases} \quad (6)$$

We can obtain the value of $c_{0,0}$ by imposing the normalization condition

$$1 = \sum_{i=0}^m \sum_{j=0}^{W_i-1} c_{i,j} = c_{0,0} \sum_{i=0}^m P_c^i \frac{W_i + 1}{2} = \frac{c_{0,0}}{2} \left[W_0 \sum_{i=0}^m (PF \cdot P_c)^i + \sum_{i=0}^m P_c^i \right]. \quad (7)$$

Now, we can obtain ξ , which is the probability that a station transmits in a slot time. As any transmission occurs when the backoff size is equal to zero, we have

$$\xi = \sum_{i=0}^m c_{i,0} = \frac{c_{0,0}}{1 - P_c}. \quad (8)$$

Let P_t be the probability that there is at least one transmission in a slot time and let P_s be the probability that a frame is transmitted successfully. Once ξ is known, P_t and P_s can be obtained as follows:

$$P_t = 1 - (1 - \xi)^n. \quad (9)$$

$$P_s = \frac{n\xi(1 - \xi)^{n-1}}{P_t} = \frac{n\xi(1 - \xi)^{n-1}}{1 - (1 - \xi)^n}. \quad (10)$$

We let L_f be the average amount of payload information successfully transmitted in a slot time and L_s be the length of a renewal interval. Finally, we can determine γ , defined by

$$\gamma = \frac{L_f}{L_s} = \frac{P_s P_t L_p}{(1 - P_t)S_t + P_s P_t T_s + (1 - P_s)P_t T_c}. \quad (11)$$

where L_p is the average frame length and S_t is the size of a slot time. Moreover, T_s is the average time that the channel is sensed busy due to a successful transmission and T_c is the average time that channel is sensed busy by the stations during a collision. We assumed for simplicity that all the stations use the RTS/CTS access method for all the transmitted frames over the IEEE 802.11 networks. In such a case collisions can occur only on RTS frames. We obtained values for the throughput under various system parameters. The analysis results were obtained using the OFDM system parameters.

In the proposed algorithm, according to the PF value during the backoff procedure, the throughput can be affected by the PF . According to the traffic type, we applied the PF value to the CW size in the above equations. The PF values for the real-time in an idle and a busy state are denoted by w_1 and w_2 , respectively. Additionally, the PF value for the non-real-time traffic in a busy state is denoted by w_3 . We considered two cases: case 1 ($w_1=1$, $w_2=3$ and $w_3=2$) and case 2 ($w_1=1.5$, $w_2=2.5$ and $w_3=2$).

Fig. 2 shows the throughput theoretically achievable by the backoff algorithm in both the cases of the IEEE 802.11e and proposed algorithm. From this figure, we can see that the proposed algorithm performs better than the IEEE 802.11e under most traffic loads. In Fig 2(a) and (b), we changed the backoff algorithm at $\phi=6$ and $\phi=7$, respectively. Around these points, the performance of the proposed algorithm is partially lower than the IEEE 802.11e. However, we can see that the proposed algorithm using threshold and PF values performs better than the conventional backoff algorithm under heavy traffic loads.

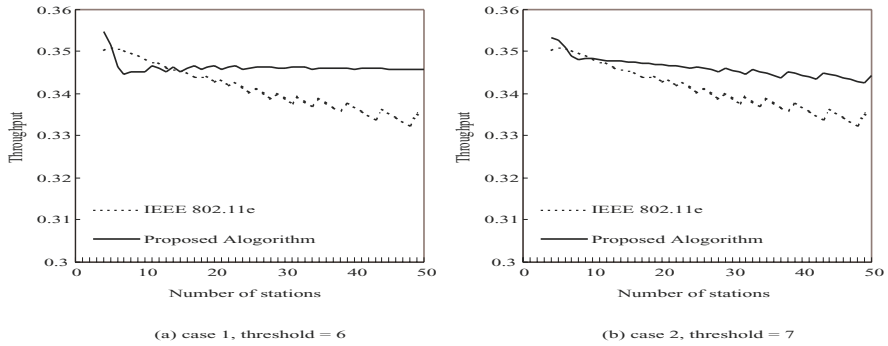


Fig. 2. Throughput under different numbers of stations

5 Conclusions

In this paper, we proposed a new backoff algorithm to enhance the throughput in IEEE 802.11e WLANs. Our basic idea is to resize the contention window under a pre-defined threshold in QBSS. We developed a Markov model to compare the proposed algorithm and the IEEE 802.11e for backoff time. We then evaluated the throughput for the proposed algorithm and IEEE 802.11e backoff algorithm. Numerical results demonstrate that the proposed algorithm performs better than the conventional IEEE 802.11e standard with EDCF in WLAN.

References

1. G. Bianchi.: Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, Mar. (2000) 535–547
2. S. T. Sheu, T. F. Sheu.: A bandwidth allocation/sharing/extension protocol for multimedia over IEEE 802.11 ad hoc wireless LANs. *IEEE J. Select. Areas Commun.*, vol. 19, no. 10, Oct. (2001) 2065–2080
3. H. Wu, S. Cheng, Y. Peng, K. Long and J. Ma.: IEEE 802.11 Distributed coordination function(DCF): Analysis and enhancement. *Proc. ICC 2002*, no. 1, Apr. (2002) 605–609
4. IEEE Standard for Wireless Medium Access Control and Physical Layer Specifications, Aug. (1999)
5. IEEE Std 802.11e/D2.0, Draft Supplement to Part 11: Wireless Medium Access Control and Physical Layer Specifications, Medium Access Control Enhancements for Quality of Service, Nov. (2001)
6. Broadband Radio Access Networks HIPERLAN Type 2, Apr. (2000)

A Light Protocol for Distance Estimation in Bluetooth Mobile Ad-Hoc Networks

Felipe Gil-Castiñeira and Francisco Javier González-Castaño

Departamento de Ingeniería Telemática, Universidad de Vigo, ETSI
Telecomunicación, Campus, 36200 Vigo, Spain
{javier,xil}@det.uvigo.es

Abstract. According to the results in this paper, Bluetooth MANET route hop count depends linearly on the distance between origin and destination, and therefore hop count may be a valid metric for MANET guidance services. However, scatternet-based or on-demand route formation algorithms for ideal MANETs are not well suited to Bluetooth technology, since route lifespan is too short even in case of moderate user walking speeds. As a consequence, we propose a feasible light protocol to estimate route lengths, based on Bluetooth inquiry/inquiry scan states. This protocol works properly for walking speeds, and it can be used to find persons in large spaces.

1 Introduction

1.1 Motivation

Nowadays, many commercial terminals have Bluetooth modems embedded. As a consequence, Bluetooth has been considered a suitable MANET-supporting technology [1]. However, as we will see in section 2, scatternet-based or on-demand route formation algorithms for ideal MANETs are not well suited to Bluetooth technology, since route lifespan is too short even in case of moderate user walking speeds.

This paper focuses on the problem of estimating distances between MANET terminals moving at walking speeds (~ 0.8 meters per second) as a function of route hop count. Our results suggest that Bluetooth MANET route hop count is a linear function of distance, and therefore hop count may be a valid metric for MANET guidance services. We propose a feasible light protocol to estimate route lengths, based on Bluetooth inquiry/inquiry scan states. This protocol works properly for walking speeds, and it can be used to find persons in large spaces.

2 Initial Approaches

All approaches in this paper are based on the same idea: a requesting device sends a “search request” with the Bluetooth address of the target device. Every node in

the scenario must participate as an intermediate node if required, depending on ongoing requests. An underlying protocol routes the request to the target device, which generates a response. Once the response arrives to the origin (requesting device), it is possible to estimate the distance to the destination (target device) from the number of route hops. The destination can send multiple responses, so that the origin can find the destination by following those directions that lead to a hop decrease.

2.1 Scatternet-Oriented Approach

The first approach consists of creating a scatternet comprising as many nodes in the scenario as possible and implementing a routing protocol over it. In a mobile scenario, a suitable routing choice is the well-known DSR algorithm [3]. **Simulation results:** In order to test this approach, we modified Blueware [4] to support a modified DSR version. Blueware is a Bluetooth simulator designed to test the TSF scatternet formation algorithm [5]. In our experiment, DSR was used to create routes to estimate distances on top of a TSF scatternet. Simulation results indicate that this approach is not valid in mobile environments, because the scatternet gets easily unconnected due to user mobility. We conclude that the scatternet-oriented approach is only valid in case of static or extremely slow nodes.

2.2 On-Demand Approach

The second approach does not assume an underlying topology of Bluetooth connections. When a node issues a distance estimation request, it initiates a route generation process that creates on-demand Bluetooth connections along the route as needed.

Assuming that all nodes scan their surroundings to find their neighbors, the normal operation of the system is as follows:

- *Source node:*
 - On user demand, it builds a *search packet* containing the target address and a zero-hop value.
 - Sets temporal connections with its neighbors and sends the search packet to them.
- *Intermediate node search packet procedure:*
 - Gets incoming search packets and decides to delete (hop limit reached) or forward them.
 - In the latter case, it increases the hop counter and attaches its address to the search packet.
 - Sets temporal connections with its neighbors and sends the search packet to them (but to those already belonging to the incoming route).
- *Target node:*
 - Gets incoming search packets and checks if previous copies have already arrived.

- If not, it reads the route in the search packet and builds a *response packet*.
 - Sets up a connection with the last node in the route and sends the response packet.
 - Resends the response packet a number of times. This is *i*) to increase the probability of a distance estimation packet reaching the source node and *ii*) to let the source node track target location changes for a while.
- *Intermediate node response packet procedure:*
- Gets the response packet and reads the next route hop.
 - Sets up a connection with the corresponding route node and passes the response packet to it.
 - If the connection is not possible, the intermediate node broadcasts the response packet to its neighbors.

Simulation results: We implemented a test-bed based on Bluehoc-ex [6] to evaluate the on-demand approach. We considered a scenario with 50 nodes in a 50×50 m² room, with initial random uniform positioning. Once the simulation starts, the nodes move along straight lines at constant speeds (with random direction changes). We observed the following:

- The variation of the distance between source and target and the variation of the number of hops in between have the same sign in most cases. For a given source and destination separated by more than 30 m, we generated a bundle of 32 source displacements (angular increments of 45° and lengths in $\{5, 10, 15, 20\}$ m). The signs of both variations (distance and hops) did not differ in 83% of the cases.
- We observed that the first packet received by the target followed a near-shortest-path route. Figure 1 shows average number of hops versus distance in meters. Note that the number of hops grows linearly with distance after a 5-m initial value.
- The density of nodes does not influence the relationship in Figure 1 significantly for more than ~ 20 Bluetooth class 2 nodes. To arrive to this conclusion, we performed a series of simulations incrementing the number of nodes in $(0, 200]$ in 5-node steps. In all cases, initial node location was random and uniformly distributed. It could be expected that the number of hops should grow with the density of nodes. However, Bluetooth nodes do not set connections depending on signal strength (as IEEE 802.11 does) and, consequently, the average number of hops in the shortest routes is similar regardless of the number of nodes, once a certain density threshold is reached.

Nevertheless, we found two important limitations of the on-demand approach:

- A long delay is due to the lack of synchronization among nodes. If the distance between source and target is ~ 10 hops, the time to receive a response ranges from 40 to 150 seconds. Although the distance covered by 10 hops can be up to 100 meters with class 2 modems, this response time is unacceptable compared to context change rates due to walking speed.

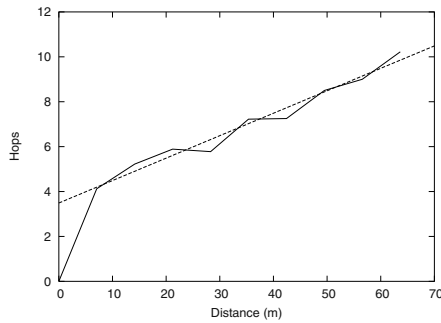


Fig. 1. Number of hops vs. distance

- For moderate to high walking speeds (i.e above 0.5 m/s) the response routes always fail. This compromises the feasibility of the system described in [1] (a system to route data in an ad-hoc Bluetooth scatternet) in mobile environments.

3 Inquiry-Oriented Approach

The long delay in the on-demand approach is due to state transitions and queue managing overload. In ideal conditions, hop delay would be the time to set a connection (1.804 seconds according to [8]). In other words, 10-hop route establishment would be ~ 36 seconds (the minimum time observed in our realistic simulations).

A Bluetooth device looking for neighbors sends BT_ID inquiry packets [2]. All devices in inquiry scan state within emitter range and listening in the same frequency will receive those packets, and they will answer with a FHS packet [2]. A BT_ID packet does not contain information on the source. It carries a 64-bit *inquiry access code* to indicate the kind of device that must answer (there is a generic code (GIAC) and a group of specific codes (DIAC). The remaining codes are reserved for future use).

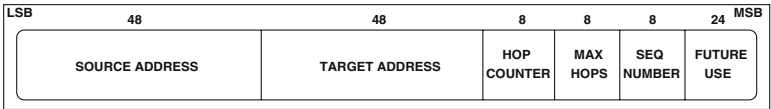


Fig. 2. Distance estimation FHS packet

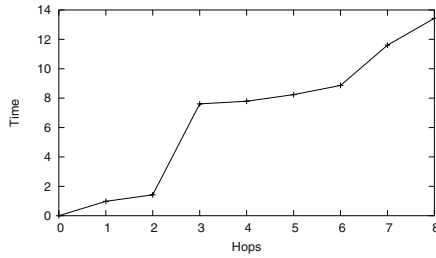


Fig. 3. Inquiry-oriented approach elapsed times

3.1 Distance Estimation Protocol

In order to overcome the limitations of Bluetooth technology, we do not establish routes. Our distance estimation protocol, based on the inquiry procedure, is described next. Its packets carry a *target address* (48 bits), a *source address* (48 bits) and a *hop counter*.

The source node simply submits a search packet to its neighbors with hop counter set to 0. Any intermediate node receiving a search packet increases the hop counter and resubmits the packet, unless a hop limit is reached (in that case, the intermediate node discards the packet). Eventually, if there are no isolated regions due to node ranges, the search packet reaches the target. The target sets the hop counter to 0 and starts the process again with another search packet swapping previous source and target addresses. When the new search packet arrives to the source node, the source node recognizes its address and obtains an estimation of the distance to the target.

Although this protocol seems extremely simple, its implementation is not obvious: evidently, it can be programmed with DM packets at application level [2]. However, DM packets require an existing connection, and we have seen in section 2 that connection setting times are unacceptable.

Alternatively, we reserve a BT_ID inquiry code for a “distance estimation inquiry”. Every participating node must alternate inquiry and inquiry scan states. While in inquiry state, the nodes send distance estimation inquiries. All devices nearby in inquiry scan state with a pending transmission (initial source transmission, target transmission or intermediate node re-transmission) must answer with a “distance estimation FHS packet” - the search packet. FHS packets have a 144-bit payload, enough for source address, target address, hop counter, sequence number and control information (hop limit, etc.). Figure 2 shows the format of the FHS search packet.

3.2 Simulation Results

We implemented the inquiry-oriented approach as a modification of Bluehoc_ex. The simulations indicate that source-to-target transmission delay is basically the aggregation of the inquiry times of all nodes involved. We must remind the reader

that the minimum inquiry time is 1.25 ms, its recommended maximum value is 10.24 s and its average value in our scenario is 3-5 s. Figure 3 shows elapsed times to obtain a distance estimation. Compared with the results in section 2.2, the delay of the inquiry-oriented approach is one order of magnitude lower.

4 Conclusions

This paper has studied the problem of estimating distances in Bluetooth MANETs as a function of route hop counts. In our scenarios, route hop count is a linear function of distance, and therefore it may be a valid metric for guidance services.

Due to technological reasons, scatternet-based or on-demand route formation algorithms for ideal MANETs are not well suited to Bluetooth technology, since route lifespan is too short even in case of moderate user walking speeds. As a solution, we have proposed a light protocol for distance estimation requiring a minimum modification of the inquiry procedure. It can be easily implemented by modifying the baseband firmware of Bluetooth devices. Simulation results indicate that distance estimation delay is acceptable for walking speeds (i.e. ~ 15 s for ~ 100 m). Forthcoming work will study the impact of Bluetooth 1.2 on distance estimation protocol performance.

References

1. Y. Liu, M. Lee, and T. Saadawi, "A Bluetooth scatternet-route structure for multihop ad hoc networks," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 229-239, Feb. 2003.
2. Bluetooth Special Interest Group, *Specification of the Bluetooth System – Volume 1 – Core*, 2001.
3. D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The Dynamic Source Routing protocol for mobile ad hoc networks (DSR)," April 2003, work in progress. [Online]. Available: <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt>
4. G. Tan, "Blueware: Bluetooth simulator for ns." [Online]. Available: <http://nms.lcs.mit.edu/projects/blueware/blueware-866.pdf>
5. T. Godfrey and J. Guttag, "A locally coordinated scatternet scheduling algorithm," in *Proceedings of the IEEE Conference on Local Computer Networks*, 2002.
6. J.-R. Sheu, "Bluehoc extended packages for Bluetooth topology construction." [Online]. Available: http://joungh.im.ntu.edu.tw/bluehoc_ex
7. T. Salonidis, P. Bhagwat, and L. Tassiulas, "Proximity awareness and fast connection establishment in Bluetooth," in *Proc. First Annual Workshop on Mobile and Ad Hoc Networking and Computing*, 2000, pp. 141-142.
8. R. Woodings, D. Joos, T. Clifton, and C. D. Knutson, "Rapid heterogeneous connection establishment: Accelerating Bluetooth inquiry using IrDA," in *Proceedings of the Third Annual IEEE Wireless Communications and Networking Conference (WCNC '02)*, 2002. [Online]. Available: <http://citeseer.nj.nec.com/463475.html>

Access Router Discovery and Selection in Advanced Wireless Networks

N. Blefari-Melazzi¹, D. Di Sorte², M. Femminella², L. Piacentini², and G. Reali²

¹D.I.E. – University of Rome "Tor Vergata"

blefari@uniroma2.it

²D.I.E.I. – University of Perugia

{disorte, femminella, piacentini, reali}@diei.unipg.it

Abstract. Advanced mobility management is a key issue to provide nomadic users with QoS-enabled services, in the future Internet. To improve Mobile IP performance, a crucial role is played by the discovery and selection of the access router to handover to. This procedure is of great importance to enable other additional mobility-related protocols to speed up handover. This paper presents and evaluates a novel approach for carrying out this task, based on a fully distributed process, which also allows timely inter-technology handovers.

1 Introduction

IP mobility protocols enable Mobile Nodes (MNs) to execute IP layer handover between Access Routers (ARs). It is known that basic Mobile IP (MIP) protocols perform poorly, in particular for supporting QoS demanding applications. To improve them, a number of different approaches have been proposed so far: (i) micro-mobility solutions, context transfer solutions, solutions that minimize packet loss and delay. The proposed enhancements assume that the new AR to which hand over is known. However, the way to discover this information is an open research topic. In advanced mobility scenarios it is important to discover the set of potential target ARs and to select the most appropriate one before handing over. A candidate access router discovery (CARD) procedure collects information about the ARs that are candidates (CARs) for the MN's handover [1]. Then, it is possible to identify the AR that best matches MN's requirements and CARs' capabilities.

In this paper, we propose and analyze by simulations a procedure for CAR discovery and Target Access Router (TAR) selection in an all-IP network controlled by a single operator. This procedure is network-assisted, distributed, and based on multicast communications. It allows intra and inter-AR handover (which, in turn, can be both intra and inter-technology), and permits to dynamically self-construct a local (i.e., relevant to an AR) map of the wireless coverage.

This work was carried out in the framework of the FIRB Projects VICOM and PRIMO, co-financed by the Italian Ministry for Education, Higher Education and Research (MIUR).

2 The CARD Approach

Under the power-saving assumption that multi-mode terminals have a single active IP interface towards the current AR, the main steps that a CARD procedure must include are: (i) reverse address translation of the L2 identifiers (L2 IDs) of the operating Access Points (APs), learned by MN from beacons, into the IP addresses of the relevant ARs; (ii) discovery of service capabilities associated with the AR-AP pair.

Multi-mode terminals having only one interface turned on are not allowed to scan for beacons of Radio Access Technology (RATs) different from the one currently used, thus they cannot obtain information from them. We solve this problem by means of a so-called probabilistic approach whose aim is to infer about layer 2 coverage. This choice enhances the basic CARD approach that relies on beacon listening only and which is suitable only for single mode terminals and multi-mode terminals with all radio interfaces turned on.

As regards the TAR selection algorithm, executed at the current AR, it takes into account load balancing issues, received power level, and estimated values of successful handover probability between involved APs. This implies (i) substantial power saving at MNs, (ii) avoiding highly complex terminal equipment, (iii) managing critical service information only among ARs (security issues), and (iv) avoiding wireless bandwidth waste. The TAR algorithm jointly selects both the best AR in the set of CARs and the best AP among those connected to the TAR.

We consider as service capabilities the bandwidth of the link from the wireless interface of an AP to the output port of the AR.

We address the issues of reverse address translation and discovery of service capabilities by defining a procedure based on multicast transmissions in the wired network at two hierarchical levels. At the highest level, the network operator defines a multicast group (MG_{op}) including all the ARs that currently provide wireless connectivity. These ARs act as multicast hosts, whereas the functions of the multicast routing are performed by the routers in the core network. In other words, ARs are the network entities exchanging multicast information. MG_{op} is used to resolve the AR IP address starting from L2 ID through explicit queries. At the lowest hierarchical level, the i th access router, AR_i , builds up another multicast group (MGi), which includes also all ARs with a coverage area that overlaps with the one of AR_i . Clearly, we consider the coverage area of each AR as the union of the coverage areas of all APs connected to it. Such MGi is used by the AR_i to efficiently distribute information about the service capabilities of its APs to the geographically adjacent ARs. The information about the coverage is not provided to ARs in a static way (i.e., by means of manual configuration, which could be not feasible in large wireless networks), but it is dynamically learned on the basis of (i) the knowledge of the L2 connectivity from MNs, and (ii) the success of previous handover events. This information is maintained (and updated) in tables locally stored at the ARs, and described below. In summary, the procedure allows associating the geographical proximity of an AR with the participation to a given multicast group. The advantage of such a procedure is that it is able to automatically self-construct this geographical mapping and also to react to variations of the coverage (e.g., the activation/deactivation of APs). The entire process is distributed and based on local exchange of update messages. For more details on such a procedure, readers should refer to [3].

A CARD table related to a given AR reports the following information: (a) the L2 IDs and the RAT identifier of the APs connected to the AR; (b) the relevant service capabilities; (c) a statistic parameter which provides a sort of coverage information at layer 2; (d) the L2 ID and the RAT identifier of each geographically adjacent AP; (e) the IP address of the relevant ARs; (f) the associated service capabilities and the parameter reporting the probability of successfully completing an handover.

3 The TAR Selection Algorithm

The TAR selection may be triggered by three events:

1. upon explicit request from the customer;
2. periodically, decided by the current AR, with period T_{TAR} , which can be adjusted by the AR dynamically, on the basis of the current load, with the aim of balancing it over the APs. In particular, each MN is assigned a period equal to

$$T_{TAR} = \max \left\{ T_D \cdot \left(1 - e^{-\delta \frac{SC}{C}} \right), T_{TAR, \min} \right\} \quad (1)$$

where SC is the amount of bandwidth available in the link from the wireless interface of the AP to the output port of the relevant AR, C is the capacity of the AP, T_D and δ are design parameters. The higher the available network resources, the longer the time period T_{TAR} . The value of T_{TAR} is firstly determined at connection setup, then it is updated at each TAR event according to the current load of the AR-AP pair the MN is attached to. When the amount of traffic increases, the frequency of TAR events increases as well;

3. upon explicit request from MN, when it detects that the received power level PW is rapidly decreasing. In particular, this request is sent when such a power is below a threshold PW_{opt} . We consider the MN outside the coverage area when PW gets below the receiver sensitivity, PW_{\min} .

When one of the events listed above occurs, since the handover is managed by the AR, the MN is required to send its handover preferences to the current AR. Two types of information are sent as preferences: (i) which types of RAT the MN is able to hand over to; (ii) the L2 IDs which have been received by the MN during the latest T_{beacon} seconds. At this point, the AR has all the needed information to run the TAR. The proposed TAR algorithm aims to select a subset of maximum N CARs, according to a score standing, computed by means of a metric. The current AR notifies the MN of the list of the ARs (with the relevant APs) and the MN can try the handover.

The metric represents the criterion that provides the quality measure, $M_{TAR}(AP_{h,s}, AP_{z,k})$, of an expected handover from the current $AP_{h,s}$ to a target $AP_{z,k}$. TAR driven handovers may happen only towards the APs which have an entry in the tables managed by the current AR. The score is associated with a candidate AR_z - $AP_{z,k}$ pair for a handover from the $AP_{h,s}$. It ranges in the interval $[0, 1]$, and its expression is given by multiplying the following three functions:

$$f_1(SC_{z,k}) = \begin{cases} \min \left\{ 1, e^{\beta \left(\frac{SC_{z,k}}{C} \right)} - 1 \right\} & \text{if } SC_{z,k} \geq B \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$f_2(PW_{z,k}) = \begin{cases} 1 - e^{-\gamma \left(\frac{PW_{z,k} - PW_{\min}}{P_T - PW_{z,k}} \right)} & \text{if } PW_{z,k} > PW_{\min} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$f_3(Pr_{AP(h,s) \rightarrow AP(z,k)}) = Pr_{AP(h,s) \rightarrow AP(z,k)} = N_{HO,SUCC} / N_{HO,TOT} \quad (4)$$

With reference to relation (2), B represents the MN bandwidth demand to satisfactorily support the current communication session, $SC_{z,k}$ is the amount of available bandwidth in the link from the wireless interface of $AP_{z,k}$ to the output port of the relevant AR, C is the capacity of $AP_{z,k}$, and β is a design parameter. The higher the value of β , the higher the score associated with the available bandwidth (SC), normalized by the capacity of the AP considered. In addition, if the network access (i.e., the new AR-AP pair) cannot accommodate the new flow, its score is zero. With reference to relation (3), P_T is the standard value of the transmission power associated with a RAT, $PW_{z,k}$ is the received power from the $AP_{z,k}$, and γ is a design parameter similar to β . Clearly, the higher the value of γ , the higher the sensitivity of the score function to power levels below PW_{opt} . With reference to relation (4), $Pr_{AP(h,s) \rightarrow AP(z,k)}$ is the estimation of the handover success probability from $AP_{h,s}$ to $AP_{z,k}$. This probability value should approximate the percentage of the $AP_{h,s}$ coverage area which overlaps the $AP_{z,k}$ coverage area. It is worth noting that, for APs of the RAT currently used, f_3 is always set to 1, while, in the case of different RATs, f_2 is always set to 1.

The expression of the failure probability of the whole CARD/TAR process is

$$Pr_F = Pr_{F1} + Pr_{F2} = (1 - Pr_E) \cdot (1 - Pr_N) + Pr_E \cdot Pr_{Rec} \quad (5)$$

where Pr_E is the probability that the TAR process returns an empty TAR list, Pr_N is the probability to successfully hand over within the N th attempt, and Pr_{Rec} is the probability that the MN, out of the coverage of the previous AP, finds another overlooked AP to hand over to by means of self-reconfiguration, i.e., without the assistance from the network, thus performing a plain MIP handover.

4 Simulation Results

In this section, we present some simulation results showing the effectiveness of the proposed CARD/TAR approach. More details can be found in [3].

The network topology consists of a number of APs and ARs. We envision the presence of three different RATs, the capacities of which are $C_1=10$ Mbps, $C_2=5$ Mbps, $C_3=2$ Mbps, respectively. The network area is a square with side of 150 meters. We have run simulations over two different coverage maps. In the first one (sparse coverage), all the area is covered by at least one RAT. In the second network

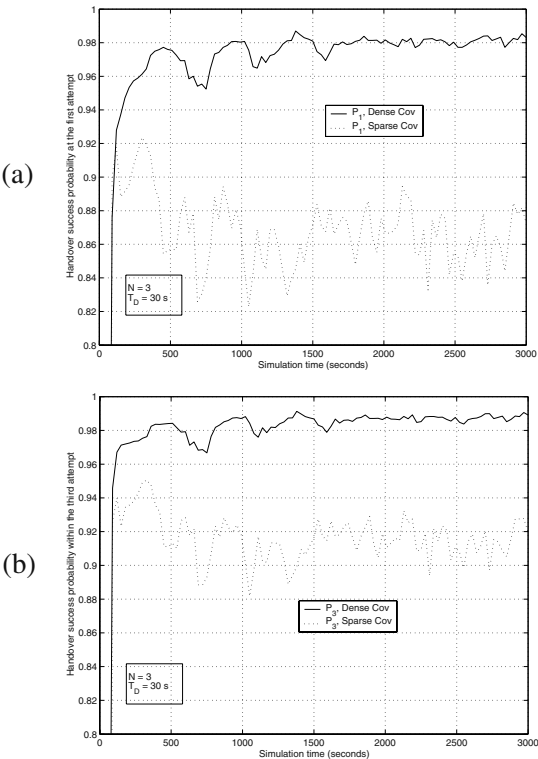


Fig. 1. Successful handover probability in both coverage cases: (a) at the first attempt; (b) by the third attempt.

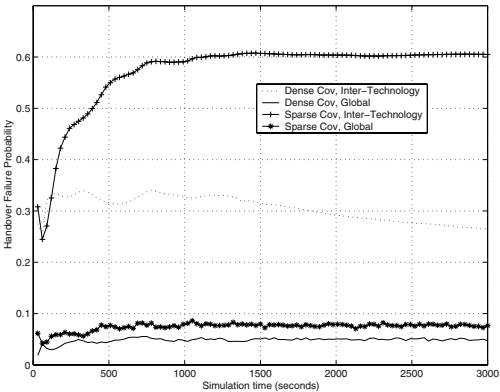


Fig. 2. Failure probability of the CARD/TAR procedure and of inter-technology handover attempts: dense and sparse coverage cases.

configuration (dense coverage) the whole area is covered by at least one AP for each RAT. Each call is associated with a bandwidth demand equal to 128 Kbps. In order to limit the number of calls rejected and handovers failed due to lack of resources, the amount of offered traffic is equal to 30% of the total capacity. The mobility model used to simulate the MNs position is the Gauss-Markov one [2]. For what concerns the CARD/TAR parameters we have made the following choices: $N=3$, $T_d=30$ s, $T_{beacon}=0.5$ s, $\beta=0.866$, $\gamma_1=3.28 \cdot 10^8$, $\gamma_2=1.46 \cdot 10^8$, and $\gamma_3=1.8 \cdot 10^8$, for RAT_{*i*}.

Fig. 1 shows the probability to successfully hand over at the first and by third attempt in the dense and sparse coverage. It is worth noting that the estimated values of probability are associated with a 95% confidence interval, not shown in figures to improve their clearness. We can see that, after a brief transient period needed for self-constructing the CARD tables, the probabilities of successfully handing over to the best-scored CARs rapidly converges to values close to 1. Clearly, the probability to successfully hand over increases with the number of attempts. We remark that performance of the CARD/TAR procedure is very satisfying for the dense wireless coverage. This is also confirmed by the global failure probability of the CARD/TAR procedure versus simulation time that is plotted in Fig. 2 together with the failure probability of inter-technology handover attempts for both coverage cases. This result was expected for two main reasons. Firstly, inter-technology handovers (the most critical ones) are not that frequent in the dense coverage case, since each RAT covers the whole area and the offered traffic load is not heavy. This means that inter-technology handovers are not solicited by load balancing actions. In fact, they are 6.24% of the total number of handovers. Secondly, in the dense coverage case, the overlapping of coverage areas of different RATs is large enough to keep the failure probability of inter-technology handover attempts low enough (around 5%, as shown in Fig. 2) to not degrade the performance of the procedure. On a total number of successful handovers equal to 164182, 128574 of them (equal to 78.31% of the total) are driven by CARD/TAR procedure. In any case, a non-zero value of the failure probability is expected, due to the intrinsic characteristic of the procedure, which, for inter-technology handovers, is blind with respect to the movement direction of driven MNs. In this regard, Fig. 2 also shows the probability to fail an inter-technology handover attempt, which results to be about 26%. As regards the sparse coverage case, the percentage of inter-technology handovers (equal to 13.37%) and the probability to fail them increase with respect to the dense coverage case (see Fig. 2). This is due to a lower number of APs and, then, to a minor overlapping of coverage areas between different RATs. The consequence is a decrease of driven handovers (equal to 68.92% on a total number of 98651) and an increase of the probability of CARD/TAR failure, which is now approximately 8%, as shown in Fig. 2.

References

1. M. Liebsch, A. Singh, H. Chaskar, D. Funato, E. Shim, "Candidate access router discovery", *Internet Draft*, work in progress, June 2003.
2. T. Camp, J. Boleng, V. Davies, "A survey of mobility models for ad hoc network research", *Wireless Communication & Mobile Computing*, 2(5), pp. 483–502, 2002.
3. D. Di Sorte, M. Femminella, L. Piacentini, G. Reali, "Target access router selection in advanced mobility scenarios", Technical Report, D.I.E.I., University of Perugia, available at the URL: <<http://conan.diei.unipg.it/mtn/personali/techrep.pdf>>.

Enhancing HIPERLAN/2 Security Aspects

Josep L. Ferrer-Gomila, Guillem Femenias, and Magdalena Payeras-Capellà*

Departament de Matemàtiques i Informàtica, Universitat de les Illes Balears
Cra. de Valldemossa km. 7,5, Palma de Mallorca 07122, SPAIN
{dijjfg,guillem.femenias,magdalena.payeras}@clust.uib.es

Abstract. Security services are not contemplated in an appropriate way in the current standards for broadband wireless LANs, namely IEEE 802.11 and HIPERLAN/2. A wrong design of the security architecture, a bad election of cryptographic algorithms and a lack of scalability, are among the criticism that these standards have received. In this paper, the security architecture adopted within the Spanish ICT (Information and Communication Technologies) project DARWIN (Demonstrator of an Adaptive and Reconfigurable Wireless IP Network) is presented. The DARWIN approach, adopting HIPERLAN/2 as a model, incorporates all the basic security services, with a high degree of flexibility and scalability, and correcting some faults of HIPERLAN/2.

1 Introduction

The striking growth of electronic data traffic, the increasing popularity of multimedia applications, and the converging trend of wireless communications and Internet technology, have spurred the evolution from second generation (2G) to third generation (3G) mobile networks and have brought new WLAN standards, like IEEE 802.11 [3] and HIPERLAN/2 [4, 5]. Within the Spanish ICT initiative, project DARWIN is working towards the definition of a flexible broadband WLAN radio access system, based on an IP network platform.

In order to fulfil the requirements on security dictated by higher layers, according to the top down approach adopted within the project, the DARWIN LC (Link Control) layer incorporates some security functions. The principal services that have been considered in the definition of the security architecture are: confidentiality, integrity, access control and authentication (mutual or unidirectional). The security architecture of DARWIN is based on the analysis of the corresponding architecture of HIPERLAN/2, and tries to solve the problems that we have identified in this standard. SSL [1] has been taken as a reference because it's a good security protocol without flaws [2].

First problem to be solved in wireless networks is access control and/or authentication. To achieve mutual authentication, HIPERLAN/2 has two methods: pre-shared key or RSA based authentication. But, it only allows the use of RSA

* This work has been supported in part by the MEC, Spain, and FEDER, under grant TIC2001-0287 and in part by the CAIB under grant PRDIB-2002GC3-18

with three possible key sizes (512, 768 and 1024 bits). This means that if the necessity of using longer keys is observed, HIPERLAN/2 will have to be redefined. We extend the use of RSA to any key size. Besides, HIPERLAN/2 does not foresee the incorporation of the public key infrastructure (PKI) based on digital certificates. We think that in broadband wireless networks, PKI can be deployed in the same way than in wired networks.

A second optional service to be provided is confidentiality. In HIPERLAN/2, this service, when desired, is activated before carrying out the authentication process, and it is not appropriate [6]. The procedure must first begin with the authentication of the parties involved in a communication and then proceed to the key exchange process.

Another problem to be solved in HIPERLAN/2 is integrity/authentication during data transfer phase. If confidentiality service is not activated, data transfer is totally unprotected in HIPERLAN/2. It means that even after a successful authentication phase, the WLAN can be attacked by non authorized users, modifying, altering or inserting data in the wireless network. We add an integrity service, independent of confidentiality service, to solve the described problem.

In this paper we present a proposal that corrects the defects of HIPERLAN/2. DARWIN proposal allows to negotiate the three basic security services: authentication, integrity and confidentiality. The design of the messages, that must be exchanged between the mobile terminal (MT) and the access point (AP), allows to incorporate new algorithms and key sizes without having to redefine the proposal. Anyway, the chosen algorithms are strong enough.

2 Security Architecture in DARWIN

The scope of DARWIN is limited to the lower layers of the OSI reference model: physical and link layers. The link layer distributes its tasks between two sublayers: MAC and LC sublayers. Furthermore, in the LC sublayer we can find a user plane (DLC, Data Link Control) and a control plane (RLC, Radio Link Control). In the control plane three differentiated modules can be found: RRC (Radio Resource Control), AC (Association Control) and DLC Connection Control. In this context, the security functions have been incorporated in the AC module of the RLC.

In order to establish the schedule of incorporation of security services, we have defined five phases: handshake, authentication, key exchange, integrity and confidentiality. Some security functions, like key refresh, security in the handover, etc., will not be described in this paper, even though DARWIN approaches them.

We will use the following notation:

- x, y concatenation of information x and y
- $H_f(m)$ hashing of message m with f algorithm
- $PR_i(m)$ encryption of message m with the private key of i
- $PU_j(m)$ encryption of message m with the public key of j
- $E_k(m)$ encryption of message m with the secret key k

3 Handshake

In the establishment stage it is necessary to negotiate which services and algorithms will be used. Related to security services, the MT carries out a proposal and the AP decides which services and algorithms will be used:

- 1.- MT \leftarrow AP: services-mt, alg-list
- 2.- AP \leftarrow MT: services-ap, alg-sel

The argument *services-mt* contains information (one octet) about the desired security services, and must be interpreted as follows (b_0 is the less significant bit):

- b_0 value 1 \Rightarrow MT requests the authentication and integrity services
- b_1 value 1 \Rightarrow MT requests the confidentiality service
- b_2 value 1 \Rightarrow MT has a public key certificate
- b_3 value 1 \Rightarrow MT wants the public key certificate of the AP
- $b_4 - b_7$ future use

If the MT demands the authentication service, automatically it demands the integrity service. This way we are not exposed to impersonation attacks during data transfer. On the other hand, if the MT and/or the AP want confidentiality, previously they have to be authenticated. By definition confidentiality means that information only is made available to true (authenticated) users.

The argument *alg-list* is an ordered list of algorithms for authentication/key exchange (RSA or pre-shared key), confidentiality (DES, 3DES, IDEA, AES, RC2 or RC4) and integrity (MD5 or SHA). In the future the number of algorithms for each service can be enlarged, without redefining the protocol messages. This feature provides the scalability property to our proposal. With RSA, parties may have to send a public key certificate. In some cases the encryption will be carried out in stream, while in other cases it will be carried out in chained blocks.

Regarding the response message sent by the AP, the byte *services-ap* contains information about the agreed security services. It must be interpreted as follows:

- b_0 value 1 \Rightarrow the authentication and integrity services are activated
- b_1 value 1 \Rightarrow the confidentiality service is activated
- b_2 value 1 \Rightarrow the MT has to send a public key certificate
- b_3 value 1 \Rightarrow the AP will send its public key certificate
- $b_4 - b_7$ future use

The argument *alg-sel* contains the algorithms that have been selected by the AP. If the AP cannot accept any option of *alg-list*, the association should be rejected.

4 Authentication and Secret Exchange

Besides the option of not using authentication, two possible authentication methods can be used in DARWIN: pre-shared key or RSA. The exchange is as follows:

- 1.- MT \Rightarrow AP: id_type, id, [certificate-mt]
- 2.- AP \Rightarrow MT: chall₁, rand-ap, [certificate-ap]
- 3.- MT \Rightarrow AP: chall₂, rand-mt, response₁
- 4.- AP \Rightarrow MT: response₂

In the first message the MT indicates the type of authentication key identifier, and the value of that identifier. The AP can use this identifier to recover the necessary key for this access. Optionally, the MT and the AP have to send public key certificates. The MT should obtain the authentication key of the AP, from the public key certificate of the AP or using the AP identifier sent in the broadcast channels. The arguments *chall₁* and *chall₂* are two random values and they are used by the other part to formulate a response to that challenge. The arguments *rand-mt* and *rand-ap* are also random values, that will be used in the generation of keys and vectors. If the responses to the challenges are correct and, thus, the authentication process ends successfully (parties are authenticated), then the association process can continue. Otherwise, the DLC (Data Link Control) connection should be rejected.

If the pre-shared key mechanism has been agreed, the responses to the challenges are computed using a hashed message authentication code, that is, a hash function with a secret parameter. In DARWIN we use the same function as in HIPERLAN/2:

$$HMAC - MD5_k(m) = H_{MD5}((k \oplus opad), H_{MD5}((k \oplus ipad), m))$$

where the input message is m , k is the secret parameter, *opad* is the character 0x5c repeated 64 times and *ipad* is the character 0x36 repeated 64 times. The values of m are as follows:

$$\begin{aligned} \text{response}_1 &= HMAC-MD5_K(\text{chall}_1, \text{rand-ap}, [PU_{MT}, PU_{AP}], \text{alg_list}, \text{alg_sel}) \\ \text{response}_2 &= HMAC-MD5_K(\text{chall}_2, \text{rand-ap}, [PU_{MT}, PU_{AP}], \text{alg_list}, \text{alg_sel}) \end{aligned}$$

The optional parameters PU_{MT} and PU_{AP} are the public keys of the MT and the AP. The parameters *alg_list* and *alg_sel* are obtained in the handshake phase. K is the pre-shared secret key between the MT and the AP, with 128 bits at least. The secret exchange based on a pre-shared key, K , is as follows:

$$\begin{aligned} \text{rand-ap} &= E_K(\text{MT}, \text{random}) \\ \text{rand-mt} &= E_K(\text{AP}, \text{random}) \end{aligned}$$

The AP generates a random value of 48 bytes, *random*, and concatenates it with a MT identifier. The result is encrypted using the key shared with the MT. The MT sends the same value *random* linked with an AP identifier, encrypted with the shared key in order that the AP can verify that it has been received correctly and in a secure way.

With RSA, the responses will be calculated through the computation of a digital signature. DARWIN allows an arbitrary key size (nevertheless, it is recommended that it be between 512 and 2048 bits). The operations are as follows:

$$\begin{aligned} \text{response}_1 &= PR_{MT}(H_{MD5}(\text{chall}_1, \text{rand-ap}, [PU_{MT}, PU_{AP}], \text{alg_list}, \text{alg_sel})) \\ \text{response}_2 &= PR_{AP}(H_{MD5}(\text{chall}_2, \text{rand-mt}, [PU_{MT}, PU_{AP}], \text{alg_list}, \text{alg_sel})) \end{aligned}$$

The secret exchange with RSA is as follows:

$$\begin{aligned} \text{rand-ap} &= PU_{MT}(\text{random}) \\ \text{rand-mt} &= PU_{AP}(\text{random}) \end{aligned}$$

The AP generates a random value, *random*, and encrypts it with the public key of the MT. Then, the MT sends the same value *random* encrypted with the public key of the AP. In both cases, once the exchange has finished, each user has the necessary material to generate the session keys. The *pre-master secret* is the value *random* ($PMS = random$).

5 Key Generation

Once the MT and the AP have exchanged the *pre-master secret*, each one should generate the session keys for encryption effects (confidentiality service) and/or the keys for the integrity service. DARWIN uses different keys in the AP to MT and the MT to AP directions. So it is necessary to generate two keys and, according to the chosen encryption algorithm, two initialization vectors (*IV*).

First of all the *master secret*, *MS*, should be generated using the *pre-master secret*, and the random values, *rand-mt* and *rand-ap*:

$$\begin{aligned} MS = & H_f(PMS, H_f('A', PMS, rand-mt, rand-ap)), \\ & H_f(PMS, H_f('BB', PMS, rand-mt, rand-ap)), \\ & H_f(PMS, H_f('CCC', PMS, rand-mt, rand-ap)) \end{aligned}$$

In the previous (and in the following) expression, *f* can be MD5 or SHA. Next, the key block, *KB*, should be computed to obtain the session keys and *IV*s:

$$\begin{aligned} KB = & H_f(MS, H_f('A', MS, rand-ap, rand-mt)), \\ & H_f(MS, H_f('BB', MS, rand-ap, rand-mt)), \\ & H_f(MS, H_f('CCC', MS, rand-ap, rand-mt)), \dots \end{aligned}$$

This process has to be repeated until enough output has been generated. Then, this key material, *KB*, has to be partitioned, as necessary, in the following order: *key-MT*, *key-AP*, *IV-MT*, *IV-AP*, *IC-MT* and *IC-AP*. The extra key material will be discarded. The values *IC-MT* and *IC-AP* will be used to calculate the integrity code. As in HIPERLAN/2, it is possible that some generated key to be a weak key or semi-weak key. If it is the case, it must be discarded and the following block of *KB* must be used.

6 Integrity and Confidentiality

The encryption allows providing the confidentiality service with respect to the transmitted data, while the keyed-hash functions allow obtaining the integrity service. If one or both services are negotiated during the association or handover phase, this encryption and/or integrity code will be used immediately after the key exchange has been carried out. Messages are encrypted and with integrity protection completely (from the most significant byte, MSB, to the least significant byte, LSB), and individually. The integrity code is generated as:

$$IC = H_f(IC\text{-}sec, opad, H_f(IC\text{-}sec, ipad, seq\text{-}num, info))$$

In the previous expression, *f* can be MD5 or SHA (the one that has been agreed in the handshake phase). *IC-sec* was generated from the secret information, and the MT and the AP have their corresponding value (*IC-MT* and *IC-AP*, respectively). The field *seq-num* is the sequence number for this message. The

argument *info* is the information to be protected. Due to the characteristics of the integrity function, it can also be obtained indirectly a second service: the authenticity of the parties involved in the exchange. The encryption is carried out on the whole message, including (if agreed) the calculated integrity code.

7 Some Concluding Remarks

HIPERLAN/2 security aspects can be improved, and here we present some enhancements. We would like to achieve compatibility with HIPERLAN/2, but we think that to carry out key exchange before authentication is a mistake. So, it's very difficult (if not impossible) to achieve compatibility with HIPERLAN/2. In this sense, we see our proposal as a future evolution of HIPERLAN/2. DARWIN incorporates all the basic security services, with a high degree of flexibility, allowing to negotiate services and algorithms to be used. The model that has been adopted in DARWIN allows the scalability of security services. DARWIN can incorporate new algorithms and different key sizes, without having to redefine the proposal of standard.

As a clearly differential element regarding HIPERLAN/2, DARWIN allows the use of public key infrastructure for authentication service. MT and AP can use authentication based on certificates, but sending these certificates is not compulsory (if parties have that information previously). In relation to key exchange, besides pre-shared key, DARWIN establishes an scheme based on RSA, a very used scheme in cryptographic protocols. For confidentiality service we have chosen strong algorithms and secure key generation processes. DARWIN provides stream ciphers and block ciphers, and establishes separated keys for the two directions of the communications (MT to AP, and AP to MT). In fact, we have adopted as a model the established one in SSL, because it is a broadly analyzed protocol and it is well known that it is a good security protocol. Finally, the integrity service was not established in HIPERLAN/2. In DARWIN approach, integrity and confidentiality are independent services. It is very useful in order to detect malicious users once authentication phase has finished (specially if parties do not use the confidentiality service).

References

1. A. Frier, P. Karlton and P. Kocher. The SSL Protocol Version 3.0, <http://home.netscape.com/eng/ssl3>
2. D. Wagner and B. Schneier. Analysis of the SSL 3.0 Protocol, 2nd USENIX Workshop on Electronic Commerce, 1996.
3. IEEE Std 802.11b-1999. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE Comp. Soc., 2001.
4. ETSI TS 101 761-1 V1.3.1. Broadband Radio Access Networks; HIPERLAN/2; Data Link Control (DLC) Layer; Part 1: Basic Data Transport Functions, 2001.
5. ETSI TS 101 761-2 V1.3.1. Broadband Radio Access Networks; HIPERLAN/2; Data Link Control (DLC) Layer; Part 2: Radio Link Control (RLC) sublayer, 2002.
6. Kent and Atkinson. Security Architecture for the IP, RFC-2401, 1998.

Analytical Modeling of a Traffic Differentiation Strategy for 802.11

(Extended Abstract)

Luca Vollero and Giulio Iannello

Federico II University, Napoli, Italy,
DIS, Dipartimento di Informatica e Sistemistica,
{vollero, iannello}@unina.it

1 Introduction

Wireless communications are going to be the common way for individuals to connect with other people and to access Internet services. However, in spite of the many advantages exhibited by wireless technologies, they still have a number of drawbacks, such as poor reliability, limited throughput, scarce QoS and security support, limiting a wider applicability. An area that deserves attention is concerned with flexible mechanisms to guarantee and control the access to the communication medium under loaded conditions. Indeed, these mechanisms may represent the basis, for instance, to provide QoS guarantees to multimedia and other real-time applications, or to enforce protection from denial-of-service attacks.

In this paper we propose a modification to the MAC layer of the IEEE 802.11 standard to handle traffic differentiation in the so-called *infrastructure* configurations ([1], section 3.8). These networks of 802.11 compliant devices are characterized by multiple *mobile* stations, acting as terminal devices, that communicate through one or more *base* stations, acting as bridges for the wireless connections. Our proposal is based on introducing a selective acknowledgment schema at the base station side, which differentiates the network congestion status perceived by the mobile stations. In other words, bandwidth allocation is controlled by changing the reaction of the back-off algorithm to messages sent from stations belonging to different classes. The key aspect of our approach is that it requires modifications only to the base station, and it can be implemented at firmware level without changes at the physical layer. No change is required to the mobile stations, making the approach feasible in practice with limited effort.

To evaluate the effectiveness of the approach we extended the probabilistic model introduced by Bianchi [3] to evaluate the throughput of the 802.11 MAC layer in saturation conditions, and we used it to analyze the behavior of a modified base station with two classes of mobile stations. The analysis reveals that the traffic differentiation mechanism is effective in protecting bandwidth resources assigned to the privileged class under a wide range of situations.

2 The Modified MAC

The Distributed Coordination function (DCF) is the basic IEEE 802.11 MAC mechanism. It is a distributed algorithm for the wireless channel access, based on a CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) schema.

Before accessing the medium, the generic transmitting station senses the channel to avoid collision with other stations and waits until the channel is idle. When the channel becomes idle, it waits for a *DCF Interframe Space*, DIFS, and then starts a *back-off* timer to randomize the starting transmitting time. Finally, when the timer expires, the station starts the transmission.

All transmissions are acknowledged. The receiver, immediately after data frame reception, waits for a *Short Interframe Space* (SIFS), and then sends back an ACK frame to the transmitter. Based on the acknowledgment status, the stations change adaptively their maximum transmitting frame rate, in order to reduce the probability that two or more transmitting stations collide.

The original version of DCF supports a form of traffic differentiation relying upon a polling schema, called Point Coordination Function (PCF). In [5], the combination of DCF and PCF was proved to limit both the maximum achievable throughput and the maximum payload size that can be transmitted. Hence, to introduce better traffic differentiation mechanisms in the IEEE 802.11, it is mandatory a modification to the standard MAC protocol. As discussed in section 5, previous proposals focused on the change to the back-off behavior and on its combination with the IFSs ([7], [10], and [11]).

While these approaches may be effective in the general case, we believe they may be improved for *infrastructure* configurations, i.e. networks of 802.11 devices characterized by the presence of one or more base stations (BSs), connected to the wired network and acting as bridges for wireless communications, and multiple mobile stations (MSs), acting as terminal devices.

The inherent asymmetric organization of infrastructure configurations can be exploited to minimize the MAC modifications required to support traffic differentiation. The approach we propose is to modify only the BS acknowledgment behavior to differentiate the network congestion status perceived by the mobile stations, saving all the other characteristics of the standard. The idea behind this proposal is that the throughput of the generic MS can be limited through this simple selective acknowledgment mechanism, allowing for a centralized control of traffic differentiation through congestion simulation.

In practice, with two traffic classes, the mechanism can be implemented by introducing a *dropping factor* d_f for the lower priority class over a control window of L received frames. Dropping $l = d_f \cdot L$ frames every L frames received from lower priority stations, causes DCF to reduce the throughput of the low priority class in order to adapt to the higher perceived congestion status.

This approach leads up to a number of advantages. First, the exploitation of the inherent asymmetric behavior of BS configurations allows us to limit MAC layer modifications to just one station. Second, the selective acknowledgment can be implemented at the firmware level without hardware changes. Third, the ability to control the station resources through a centralized mechanism allows us

to verify continuously the total amount of available bandwidth and to easy define mechanisms for the stations admission control and resources redistribution.

3 The Analytical Model

In [3], Bianchi presents an analytical model for the evaluation of the IEEE 802.11 LAN throughput in saturation conditions. The model assumes that the system reaches a stable state where the frame collision probability can be supposed constant and independent of other parameters. The mathematical results derived under these hypotheses are compared with performance data produced through simulations, confirming the effectiveness of the approach.

As a first step to evaluate the effectiveness of the proposed modified MAC, we choose to follow a similar analytical approach and extended the Bianchi's model to develop an analysis in saturation conditions.

To make the reading of the following formulas easier, we have summarized in table 1 all symbols used by our extended model.

Table 1. Symbols used in the formulas.

Name	Description
m	maximum retransmission back-off
W	minimum back-off window
n_i	number of transmitting stations in class i
τ_i	probability that a generic station in class i transmits on a given time slot
p	probability that a given frame collides with other transmitted frames
Δ	additional collision probability perceived by low priority stations

To maintain the model as simple as possible, we limited our analysis to the case of two transmission classes, referred to in the following as class 1 (higher priority) and class 2 (lower priority), respectively.

The key approximation in the Bianchi's model is that each packet collides with constant and independent probability p . Our schema uses the same approximation and modifies the model introducing two transmission classes. Since our differentiation technique corresponds to different collision probabilities, in the extended model transmission and collision probabilities are mutually related according to the following set of equations:

$$\tau_1 = \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)}$$

$$\tau_2 = \frac{2(1-2\tilde{p})}{(1-2\tilde{p})(W+1) + \tilde{p}W(1-(2\tilde{p})^m)}$$

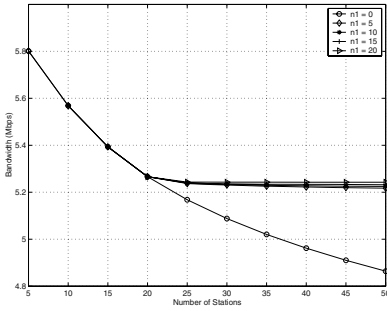


Fig. 1. Total bandwidth utilization.

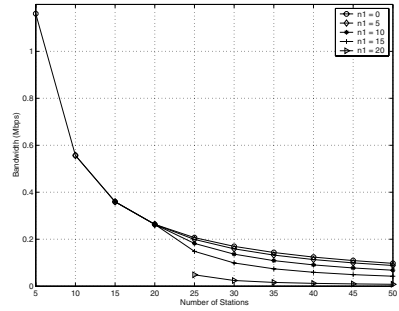


Fig. 2. Bandwidth utilization of each station in class 2.

$$p = 1 - (1 - \tau_1)^{n_1-1} (1 - \tau_2)^{n_2}$$

$$\tilde{p} = 1 - (1 - \tau_1)^{n_1} (1 - \tau_2)^{n_2-1} + \Delta$$

where τ_1 and τ_2 are the transmission probabilities, p and \tilde{p} are the collision probabilities, and n_1 and n_2 are the number of transmitting stations for traffic classes 1 and 2, respectively, and Δ is the additional collision probability induced by the modified MAC for traffic class 2.

Given Δ , the above extended model can be solved, obtaining the corresponding transmission and collision probabilities of the two classes (see [4]). With these parameters we can derive both the dropping factor d_f the BS has to apply to achieve the desired traffic differentiation, and the normalized saturation throughput of the two classes, which represents the primary performance index to evaluate the selective acknowledgment strategy.

4 Performance Characterization

In this section, we report the results obtained from the model presented above. We evaluate the behavior of a set of MSs partitioned into two traffic classes, varying both the total number of stations and the number of stations in class 1. We developed the analysis using a constant frame length of 1 Kbytes and a constant assured throughput of 250 Kbps for each MS in the privileged class. These parameters could be also varied, but for the sake of simplicity we maintained them constant (for a more detailed analysis, see [4]). For the model parameters we used the typical values of the 802.11b standard.

Total bandwidth in saturation conditions vs. the number of MSs is reported in figure 1. Different curves refer to a different number of class 1 MSs. The curve corresponding to $n_1 = 0$ represents the behavior of standard DCF according to

what estimated by Bianchi's model (the curve presented here differs from the one reported in [3] because a transmission rate of 1 Mbps is assumed in the cited article). Graphs show that the use of our differentiation technique leads to a reduction of unused bandwidth when the total number of MSs overcomes a given threshold.

The bandwidth of a single MS in class 2 is reported in figure 2. It shows that our mechanism is able to differentiate traffic effectively avoiding complete starvation of stations in low priority class.

5 Related and Future Work

In the literature, a lot of work has been done on the problem of distributed access mechanisms supporting traffic differentiation and throughput enhancement.

In [5], Visser and Zarki present simulation results for a combination of speech and data communications over a IEEE 802.11 LAN, using PCF the polling schema provided from the 802.11 standard. In [6], Sobrinho and Krishnakumar present a distributed solution supporting real-time transmissions, specialized only for isochronous sources. In [7], Deng and Chan propose a method to support station priority, changing both the interframe space (IFS) used between data frames, and the back-off mechanism.

In [10], Vaidya *et al.* present an access schema relying upon an adaptation of fair queuing ([8], [9]) to the wireless environment. In [11], Banchs and Pérez propose an extension of the IEEE 802.11 MAC protocol referred to as ARME (Assured Rate MAC Extension).

Finally, motivated by the growing interest in wireless networks supporting QoS, the IEEE 802.11 Working Group started an activity to enhance the 802.11 MAC protocol. IEEE 802.11 Task Group E is currently defining enhancements to the basic 802.11 MAC, called 802.11e ([2]). The effort aims to introduce and extend the mechanisms proposed in the literature, and in particular those based on: (i) back-off and interframe space differentiation, and (ii) multiple independent back-off instances for each MS.

While our approach, being limited to infrastructure configurations, is less general of most of the above mentioned proposal, its implementation requires very limited changes only on the base station side.

Another interesting aspect of our approach is the ability to enforce absolute guarantees. Indeed, simply varying the simulated congestion, it is possible to increase or reduce the level of traffic differentiation, or, alternatively, adapt the current level to meet specific minimal requirements. Other solutions, like those discussed in [7] and [10], do not specify how to accomplish the latter task and how the differentiation level can be controlled easily.

Finally, as stressed by the performance analysis, our mechanism allows a fine grained control over resource assignments and it leads to improved overall channel utilization.

Future works will be mainly focused on validation of the used analytical model and on characterization of system dynamic behavior by simulations. A

modified version of berkeley network simulator ([12]) has been already developed to achieve both these evaluations. Another future work is the extension of the analytical model to more general and complex cases, such as considering more than two traffic classes or RTS/CTS and hybrid MAC protocols.

Acknowledgements. This work has been carried out under the financial support of the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) in the framework of the FIRB project "Middleware for advanced services over large-scale, wired-wireless distributed systems (WEB-MINDS)", and of the project "Scalability and Quality of Service in Web Systems".

References

1. IEEE 802.11 WG, "Reference number ISO/IEC 8802-11: 1999(E) IEEE Std 802.11, 1999 edition. International Standard [for] Information Technology-Telecommunications and Information exchange between system-Local and metropolitan area networks-Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", 1999.
2. IEEE 802.11 WG, *Draft Supplement to STANDARD FOR Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Services (QoS)*, IEEE 802.11e/D2.0, Nov. 2001.
3. G. Bianchi, Performance Analysis of the 802.11 Distributed Coordination Function. *IEEE Journal on Selected Area in Comm.* V18, N3, March 2000.
4. L. Vollero and G. Iannello, Analytical Modeling of a Traffic Differentiation Strategy for 802.11, Tech. Rep., Computer Science Department, Federico II University, Napoli, Italy, July 2003
5. M. A. Visser and M. El Zarki, Voice and data transmission over an 802.11 wireless network. In *Proceedings of PIMRC'95, Toronto, Canada*, pp. 648–652, September 1995.
6. J. L. Sobrinho and A. S. Krishnakumar, Real-time traffic over the IEEE 802.11 medium access control layer. *Bell Labs Technical Journal*, pp. 172–187, Autumn 1996.
7. D-J. Deng and R-S. Chang, A priority scheme for IEEE 802.11 DCF access method. *IEICE Transactions on Communications*, VE82-B, pp. 96–102 January 1999.
8. S. J. Golestani, A self-clocked fair queueing scheme for broadband applications. In *Proceedings of IEEE INFOCOM*, 1994.
9. P. Goyal, H. M. Vin, and Cheng, Start-time fair queueing: A scheduling algorithm for integrated services packet switching networks. *IEEE/ACM Transactions on Networking*, 5:690–705, October 1997.
10. N. H. Vaidya, P. Bahl, and S. Gupta. Distributed fair scheduling in a wireless LAN. In *Proceedings of ACM MOBICOM 2000*, Boston, MA, August 2000.
11. A. Banches and X. Pérez, Providing throughput guarantees in IEEE 802.11 wireless LAN. In *Proceedings of WCNC*, Vol. 1, pp. 130–138, March 2002.
12. NS2, The Network Simulator; <http://www.isi.edu/nsnam/ns/>

Author Index

- Abolhasan, Mehran 144
An, Sunshin 365
Anto, A.J. 171
Awerbuch, Baruch 253

Battiti, Roberto 285
Berndt, Hendrik 227
Blefari-Melazzi, N. 383
Bonnet, C. 301
Breyer, Tobias 43
Bruno, Raffaele 73

Cano, Juan-Carlos 29
Chaudet, Claude 101
Chen, Wen-Tsuen 158
Chessa, S. 184
Chung, Min Gyo 359
Conti, Marco 73
Coupechoux, M. 301
Cuomo, Francesca 116

Do, Hung Tuan 199
Donmez, Mehmet Yunus 315

Egeland, Geir 344
Engelstad, Paal 344
Ersoy, Cem 315

Femenias, Guillem 389
Femminella, M. 383
Ferrer-Gomila, Josep L. 389

Gangadharipalli, Sridhar 16
Gil-Castiñeira, Felipe 377
Golwelkar, Uday 16
González-Castaño, Francisco Javier 377
Gregori, Enrico 73
Guérin Lassous, Isabelle 101

Hirschfeld, Robert 227
Holmer, David 253
Hwang, Jun 359

Iannello, Giulio 395
Isik, Sinan 315

Jang, Kil-Woong 371

Jin, Xin 271

Kawamura, Katsuya 227
Kim, Seong-Lyun 241
Kim, Yong Chul 359
Kim, Yunkuk 365
Klein, Michael 43
König-Ries, Birgitta 43
Kumar, V. 301

Lee, Wei-Ting 158
Lestable, T. 301
Li, Bo 285
Lin, Chun-Hung 57
Liu, Chien-Yuan 57

Maestrini, P. 184
Manzoni, Pietro 29
Mé, Ludovic 213
Meer, Hermann de 329
Melodia, Tommaso 116
Meng, Bin 271

Narayanasamy, P. 171
Noh, Wonjong 365

Obreiter, Philipp 43
Ocampo, Roel 329
Onozato, Yoshikuni 199

Pagani, Elena 130
Payeras-Capellà, Magdalena 389
Piacentini, L. 383
Pietro, R. Di 184
Puttini, Ricardo Staciarini 213

Reali, G. 383
Rossi, Gian Paolo 130
Rubens, Herbert 253

Sanchez, Miguel 29
Sorte, D. Di 383
Sousa, Rafael Timóteo de, Jr. 213
Subramanian, Anand Prabhu 171

Tebaldi, Stefano 130

Trung, Tran Minh 241

Tsai, Tzu-Chieh 87

Tu, Chien-Ming 87

Varadarajan, Sridhar 16

Vasudevan, Janani 171

Vollero, Luca 395

Wallbaum, Michael 1

Wang, Hongbo 271

Wasch, Torsten 1

Wysocki, Tadeusz 144

Zhang, Yaoxue 271

Žerovnik, Janez 101